IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 28, 2020, accepted August 5, 2020, date of publication August 17, 2020, date of current version September 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017082

Efficient Clustering of Emails Into Spam and Ham: The Foundational Study of a Comprehensive Unsupervised Framework

ASIF KARIM, SAMI AZAM, (Member, IEEE), BHARANIDHARAN SHANMUGAM, (Member, IEEE), AND KRISHNAN KANNOORPATTI
College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia

Corresponding author: Sami Azam (sami.azam@cdu.edu.au)

This work was supported by Charles Darwin University, Australia, through the Research Training Scheme (RTP) Scholarship Program.

ABSTRACT The spread and adoption of spam emails in malicious activities like information and identity theft, malware propagation, monetary and reputational damage etc. are on the rise with increased effectiveness and diversification. Without doubt these criminal acts endanger the privacy of many users and businesses'. Several research initiatives have taken place to address the issue with no complete solution until now; and we believe an intelligent and automated methodology should be the way forward to tackle the challenges. However, till date limited studies have been conducted on the applications of purely unsupervised frameworks and algorithms in tackling the problem. To explore and investigate the possibilities, we intend to propose an anti-spam framework that fully relies on unsupervised methodologies through a multi-algorithm clustering approach. This article presents an in-depth analysis on the methodologies of the first component of the framework, examining only the domain and header related information found in email headers. A novel method of feature reduction using an ensemble of 'unsupervised' feature selection algorithms has also been investigated in this study. In addition, a comprehensive novel dataset of 100,000 records of ham and spam emails has been developed and used as the data source. Key findings are summarized as follows: ***I)*** out of six different clustering algorithms used – Spectral and K-means demonstrated acceptable performance while OPTICS projected the optimum clustering with an average of *3.5%* better efficiency than Spectral and K-means, validated through a range of validations processes ***II)*** The other three algorithms- BIRCH, HDBSCAN and K-modes, did not fare well enough. ***III)*** The average balanced accuracy for the optimum three algorithms has been found to be ≈94.91%, and ***IV)*** The proposed feature reduction framework achieved its goal with high confidence.

INDEX TERMS Machine learning, phishing attack, spam detection, spam email, spam filtering, clustering.

I. INTRODUCTION

Email spamming can be defined as the act of distributing unsolicited messages, oftentimes sent in bulk using email. Emails, sent for legitimate purposes, are known as Ham [1]. Spammers use the act of spamming for not only marketing purposes, but also to achieve more malicious goals such as reputational damage and financial disruption, both in institutional and personal front. Emails are still considered the primary choice for the scammers when comes to delivering malware. Financial gain is one of the main motivation for the spammers. Estimation is that spammers may earn around USD 3.5 million yearly from spamming [2].

By the end of 2019, there were nearly 4 billion active email accounts worldwide [3]. In fact in 2019, approximately 294 billion emails have been exchanged daily, 50% of which were just spams [4]. Needless to say, this substantial volume of spam emails circulating through a public network like internet is continually having a damaging and costly footprints on the communication bandwidth, available memory on email servers and CPU cycles, in addition millions of everyday users' time and patience in dealing with these spam emails. In a recent report, FBI stated that malicious spamming has incurred a financial damage of USD 12.5 Billion to business email consumers in 2018 [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

The United States in general is known to be the largest source of spam emails, however, in recent times other countries often outnumber USA in originating spam emails. As of April 2019, Russia and Brazil have surpassed USA and China (another notable spam email producing country), to produce approximately 14% and 16% of total volume of worldwide spam respectively [6]. Though there were legislations such as CAN-SPAM (Controlling the Assault of Non-Solicited Pornography and Marketing Act) to protect the users, it did not achieve the expected deterrent effect on the spammers [7]. World's top 70% spam gangs, responsible for coordinated worldwide spamming, have their roots in USA [2]. Meanwhile in Oceania, recent reports indicate Australian businesses and consumers already lost nearly AUD 28,375,373 due to email fraud by the end of 2019 [8].

The contributions of this research can be summarized in ***I)*** Developing a comprehensive novel database from multiple email corpuses that may be universally used for any number of related research, ***II)*** Investigating the effect of unsupervised clustering of only the domain and header information of both ham and spam emails excluding the subject header, ***III)*** Employing a robust unsupervised feature reduction algorithm for dimensionality reduction, and ***IV)*** Lay the pathway for subsequent research and development of the complete framework.

A. SCOPE AND MOTIVATION OF THE RESEARCH

A number of research initiatives in this field using supervised, and a combination of supervised and unsupervised machine learning approaches have already been undertaken and some of those are quite extensive. However, until now, very limited work has been carried out based solely on unsupervised methodologies. Moreover, the type of analysis, mostly supervised, has been found to be revolving around the subject and content of the email, but header and domain information, as well as the presence of underlying scripts within the emails have not been investigated to any great depth, especially using purely unsupervised methods. However, as we know, the problem domain of ham and spam email is not constrained only within a single facet of the email subsystem, but rather all the sub-parts need evaluation and investigation. This research of ours is a comprehensive attempt at resolving this vacuum. Collecting and managing labelled data for supervised learning algorithms is often quite complex and expensive [9], whereas unsupervised clustering can work with unlabeled data. Though it is vitally important to dissect all the parts of an email including domain, header and content while building anti-spam models [10], this study will look into the effect of unsupervised clustering of only the domain and header information of both ham and spam emails excluding the subject header.

Additionally, a key gap in many of the existing works that we have inspected, is that those do not clearly state why certain features have been preferred over other features, or why some other features have been left out altogether. We wanted to avoid such dilemma and thus came up with a novel unsupervised feature reduction mechanism that can confidently indicate the non-significant feature set that can be left out of the analysis. Some of the other studies did in fact use feature selection processes but we believe the technique developed by us provides increased confidence on the usefulness and impact of the significant features.

In future studies, the content, scripting information and the subject header will also be thoroughly examined under the umbrella of purely unsupervised methodologies. The knowledge gained from this study and the subsequent ones will be hybridised to generate the complete spam filtering solution. This study is a critical and novel undertaking as we have found no previous studies to exactly match the objectives of this research.

II. RELEVANT STUDIES

Research initiatives in the field of unsupervised clustering of emails into spam and ham purely using header and domain information are rather scant, despite that, the following section sheds light on some of the closely related works.

The framework introduced by Smadi *et al.* [11], named, 'Phishing Email Detection System (PEDS)' uses unsupervised clustering, in conjunction with both supervised and reinforcement learning techniques [12]. Such amalgamation equips the system with an enhanced capability to modify itself based on the identified modifications and changes in the environment. The proposed model analyzes a number of header and domain information such as MessageIDs, Sender's Domain, email's content class, whether the email is multi-part, number of receivers and attachments, reply address etc. The system mainly aims at tackling Zero-Day Phishing attacks [13]. Based on the environmental parameters, the heart of the system- 'Feature Evaluation and Reduction (FEaR)' algorithm, can rank and select the critical features from emails dynamically. FEaR is based on Regression Tree (RT) algorithm. Immediately after the execution of FEaR, another novel algorithm, DENNuRL (Dynamic Evolving Neural Network using Reinforcement Learning) lets the core Three-Layer Neural Network of PEDS to evolve dynamically and stitch together an optimum Neural Network. Though the achieved accuracy rate is 99.05%, some of the features employed such as 'NumLinkNonASCII', 'BodyDearWord', 'BodyNumChars', 'ContainScript', and 'BodyNumWords' are rather not that conventional as the degree of impact of these features have not been discussed. Authors have not presented the logic behind the inclusion of these, leaving scope for further analysis. Inclusion of some of the critical domain features such as source IP and age of domain have also been ignored.

A critical issue in literature and historic documents that has gained much attention is the determination or attribution of 'Authorship'. Researchers have developed strategies, such as identifying patterns pertaining to stylistic, syntactic and grammatical features available in such documents, to successfully identify and group original authors of such documents. Despite emails being highly unstructured, Alazab *et al.* [14] attempted to introduce the idea of 'Authorship Attribution' for phishing campaign identification. The authors have deployed an Unsupervised Automated Natural Cluster Ensemble (NUANCE) methodology to achieve approximate clustering of spam emails. 57 stylistic features have been used by the researchers such as, for instance-total word count of the text of the email, total count of the punctuations used in the email body, total number of contractions present in the email, total number of URLs present in the body of the email, total number of obfuscated words present in the email etc. Semantic features along with its combination with stylistic features have also been considered. Semantic features aid is explaining how words that share certain features may be members of the same domain. The eventual clusters are obtained by hierarchically clustering the approximate sets, churning out 27 different clusters. Even though the system is quite impressive and provides improved results in the general direction of 'authorship attribution' in spam campaign detection, however, the intra-dynamics (for instance spammers interchanging or borrowing different functionalities from each other) taking place within different campaign groups may often pass undetected. Though the work mainly focusses on elements found in 'Authorship Attribution' process, nevertheless, such research attempt may be positively improved by the addition of header and domain features.

In [15], the authors have developed a clustering solution to detect spam based on spamming campaigns. They have used FP-Tree (Frequent Pattern Tree) algorithm [16] to identify spam campaigns. Authors have chosen OrientDB (a NoSQL Database) to store the campaign spams. Features were extracted from these emails so that the FP-Tree can be built based on the frequency of occurrence of each of the features. Emails are then clustered into spam campaigns based on the similarity of extracted features. Several header features, for instance- content type, character set, subject etc. have been used. Features from other parts of an email have also been put to use. However, FP-Tree, used in this research for clustering purposes using different features of an email, is extra sensitive to even minor of changes in layout or feature structure. Such minimal changes will cluster spam emails from a similar campaign into two different campaigns.

In a technical report, Blanzieri and Bryl [17] discussed several aspects of different learning algorithms aimed at spam filtering. The paper highlighted a number of proposals to alter or modify email transmission protocols in a view to encompass techniques to reduce spam emails as much as possible. Some methods focused solely on message content while others combined header or subject with content.

Al-Saaidah in his thesis [18] combined both K-means clustering and other classification methodologies to increase the detection accuracy of phishing attacks [34]. Various header and domain features have been used such as subject header, To, From and Reply domain as well as the presence of suspicious java scripts. The result projected that the combination of classification and clustering provides slightly better detection result than a standalone model, attaining an accuracy of 98.37%%. The project also used automated and manual feature selection techniques.

A limited amount of work has also been done on clustering spam and ham emails based on algorithms related to Artificial Immune Systems [19], such as Negative Selection Algorithm [20] and even custom-developed Genetic algorithms [21].

Under the domain of Unsupervised methodologies, "self-supervised learning" is crucial, and has been widely used in research domains based on computer vision as well as for anomaly detection in several fields. Self-supervised learning formulates new surrogate labels artificially or extract robust feature set through the characteristics or structure of the unlabeled data itself [66]. However, the studies done on effective ham and spam email differentiation using self-supervised learning is rather scant. We did find several flavours of 'Autoencoders' (a class of Artificial Neural Network that generates efficient feature representation of inputted dataset in an unsupervised fashion [67]) have been used in some studies in a self-supervised fashion to function as part of the overall proposition (primarily supervised).

Mi *et al.* [68] in their findings have shown that Autoencoders used in a stacked ensemble can provide greater computational ability, better feature reconstruction and higher accuracy while detecting spam emails than other traditional supervised algorithms such as Support Vector Machine, Naïve Bayes, Random Forest, Multilayer Perceptron and few others. Their study is based on commonly available PU spam corpora [69]. In another study an improved variant of Stacked Autoencoders has been used to analyze spam emails in Chinese, yielding better performance than traditional methods [68]. Additionally, in their research [70], Douzi *et al.* used Autoencoders to automatically learn the hidden feature representation of URL(s) embedded within the email content in an unsupervised manner in a view to determine the possibility of phishing scams. The learnt feature representation can then be used as an input to supervised classifiers. Though the idea seems promising, but no actual experimentation results have been reported by the authors.

Martino *et al.* [75] in their paper proposed a content based multiclass spam email identification framework. Unsupervised hierarchical clustering along with some supervised classifiers in combination with TF-IDF have been used to develop the model. TF-IDF combined with SVM performed the best with an accuracy of 95.39%. However, the dataset used for training is heavily skewed towards one particular class, thus the testing needs to be done using more balanced dataset.

Fragos [76] performed a *K*-means clustering to achieve a 2-way classification (ham and spam emails). PCA was first used to lower the dimensionality of the data before the clustering steps. The results show a 'Recall' measure of 94.91% and 98.57% in detecting ham and spam emails. However, key limitation is that the solution does not produce any consistent result in each run and the authors have not clarified why some

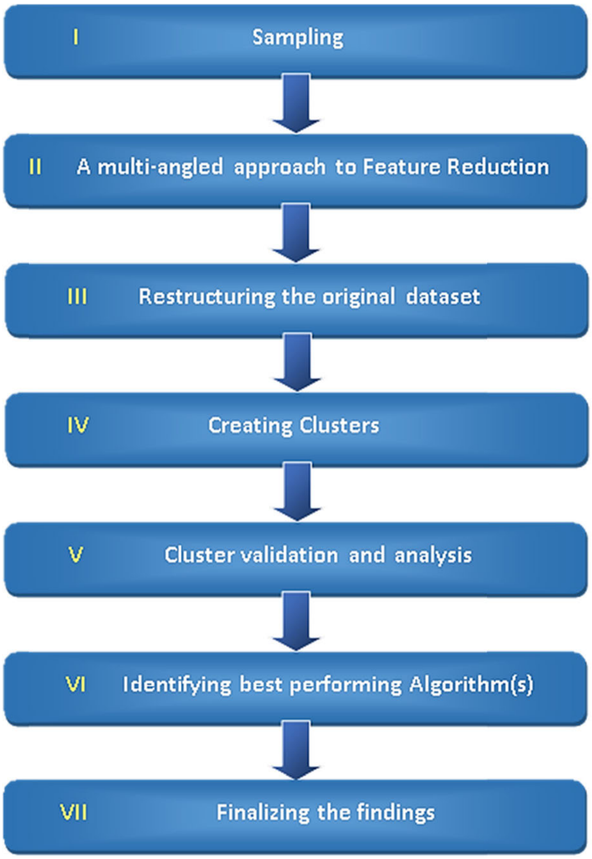

I Sampling
II A multi-angled approach to Feature Reduction
III Restructuring the original dataset
IV Creating Clusters
V Cluster validation and analysis
VI Identifying best performing Algorithm(s)
VII Finalizing the findings

FIGURE 1. An overview of the workflow.

of the features were left out while others were selected for the study.

The approach that we have taken is unique and more comprehensive especially considering the above studies.

III. OUR APPROACH AT A GLANCE

As the name suggests, unsupervised learning refers to the fact that the model will not have any labelled data to work with, and thus no training will be provided; whereas supervised approaches have the downside of training over a large amount of manually and costly tagged email corpora [27]. Now based on the dataset, unsupervised algorithms generally attempts to figure out common features within a group of items and rearranges the data points in clusters based on the commonality [22]. It is also computationally efficient and less time consuming [23], [24] than supervised approaches. Apart from the usual 'distance based' clustering [25], where a certain distance metric such as 'Euclidean Distance [26]' is applied to determine similarity between data objects, 'density based' clustering is also useful in certain domains.

As can be observed from Fig. 1, our approach comprises several steps, at step *I* (Fig. 1), we have sampled our pre-processed dataset $\boldsymbol{X}$ of 100,000 records and 10 features, storing spam and ham emails in the ratio of 2:1. Note that this novel dataset ($\boldsymbol{X}$) is completely custom built from three other publicly available datasets and we will have a brief discussion on it in section IV. The sampled dataset $\boldsymbol{S}$, houses 50,000 records such that $\boldsymbol{S} \subset \boldsymbol{X}$, where (67% spam and 33% ham) with all 10 original features intact.

The process of feature reduction is initiated at step *II* (Fig. 1), where we proposed a different approach of selecting the most impactful features, which will be discussed in due course.

Step *III* (Fig. 1) deals with restructuring original dataset to reflect the output of the previous step; in addition, the dataset was broken down into two separate parts (containing 60% and 40% of the data respectively) and used in two different runs for clustering purposes. Afterwards, different clustering algorithms were applied in step *IV* (Fig. 1) on one of the subsets holding 60% of the data. Not all unsupervised algorithms can be tuned to produce exactly two clusters, so we had to select only those which can be parameterized to do so.

Step *V* (Fig. 1) incorporates validating the clusters obtained in step IV. We employed a range of internal and external measures, to the produced clusters to confidently quantify the performance and true detection rate of the algorithms.

At this stage, in step *VI* (Fig. 1), we were in a position to identify the top performing algorithm(s). Finally, at the last step, we repeated steps IV-VI to evaluate the findings obtained in the previous step, but with the other dataset holding the remaining 40% data. The result confirms whether the best performing algorithms found earlier in the first run indeed consistently perform on email header features in clustering emails into spam and ham.

IV. THE DATASET

Though there are number of pre-processed publicly available datasets on ham and spam emails, but we had few criterion to begin with that needed to be fulfilled. Such as:

1) A dataset of both email content and all the common header fields,

2) The size of the dataset needed to be sufficiently large, around 100,000 so that more realistic performance measure and nature of clustering can be obtained, and

3) Email dataset that is not confined within a specific geographical zone.

Unfortunately, the public datasets that are already in a ready-made state such as LingSpam, Hunter SpamBase, SpamAssassin and PUA to name a few, did not fulfill the above criterion – as those were either not of expected volume, not enough information relating to header and content or particularly linked to few specific geographical area. Thus we were left with no choice other than to build such a dataset on our own from the publicly available raw and non-curated email corpuses (available as text files). We did not really use any of our own email records for this research nor plan to use in the subsequent development of our proposition.

The seminal database of over half a million records was first created from three publicly available email collections (2017 and 2018 spam collection by Bruce Guenter [30], TREC [31] and Enron dataset [32]), containing both ham and spam emails. These archives store emails in sufficient volumes in textual format, including headers. This seminal database is pivotal to the whole framework as the subsequent investigations after this research will also use this raw data source for the formation of the required pre-processed datasets. The pre-processed one used in this study, has also been created from the above raw database and has 10 features and randomly selected 100,000 records of which 67,000 are spam emails and the rest is ham. A portion of this dataset can be seen in Table 1.

WHOIS information repository has also been frequented quite heavily to populate the seminal database for certain domain based features. We have extensively used Python 3.6 to code the data extraction and engineering algorithms as well for feature reduction, clustering and validation purposes. Some R packages have been utilized for visualisation. MySQL has been deployed at backend for data storage.

The header field 'Subject' has been left out in this research as we felt it is more suitable to be coupled with the content of the email, and should best be earmarked for a separate research initiative focusing on content analysis. The below discussion briefly highlights the features used.

1. Diff_FromDomActDom: Indicates whether the domain contained in the 'From' field is different to that of the actual originating domain –extracted from the earliest 'Received' field if that field does not contain values such as 'localhost' or 'localdomain'; in that case domain mentioned in the second-earliest 'Received' field is extracted.

2. BL_IP: Identifies whether the source IP of originating domain has been 'Blacklisted' by a number of reputable spam reporting watchdogs such as *spamhaus*, *barracuda*, *sorbs* etc. to name a few.

3. Created_within_1_year: Indicates whether the originating domain has been set up within the previous 12 months from the date of the email. The mail date can be extracted from the header.

4. Expire_in_13_months: Points out whether the originating domain will expire in 13 months from its creation date. It has been reported in some experiments that spammers most commonly register spamming domains anywhere between five to around a year [28], [29].

If e is the set describing possible values for the above four features then $\mathbf{e} = \{\mathbf{0}\ \textbf{(false)},\ \mathbf{1}\textbf{(true)}\}$ as these features are Boolean in nature (stored as integers, 0 and 1).

5. Mail_dt: Date of the email sent, as mentioned before, can be extracted from email headers.

6. Cntry: Internet domain of the Country from where the email has been sourced.

7. MsgID: Message-ID of the email, it is an alphanumeric string.

8. Tm: Time of the email received.

9. Cn_typ: Content type of the email.

If $\boldsymbol{e}$ is the set describing possible values for the above five features (#5 - 9) then $\{\mathbf{e}\textbf{:}\ \mathbf{e} > \mathbf{0}\ \textbf{and}\ \mathbf{e} < \mathbf{1}\}$. These five features are Categorical and hence converted into 7-digit numeric values through SHA256 hashing.

10. Hop: A count of how many mail exchange(s) the email had to pass through before reaching the destination. The type of value for this feature is integer.

There are some header features, for instance, 'return_path', that may or may not have an impact, but we had to leave those out as only the features common to all emails generated by any mail server or email client have been included in this study.

Henceforth all the features starting from *Diff_FromDomActDom* till the last one, *hop*, will sequentially be referred to as $\boldsymbol{f_0}$, $\boldsymbol{f_1}$, $\boldsymbol{f_2}$, $\boldsymbol{f_3}$, $\boldsymbol{f_4}$, $\boldsymbol{f_5}$, $\boldsymbol{f_6}$, $\boldsymbol{f_7}$, $\boldsymbol{f_8}$, $\boldsymbol{f_9}$ respectively. *Appendix A* has more on data construction.

V. THE COMPLETE WORKFLOW IN DETAIL

In this section, the detail discussion on each of the subprocess of the complete framework will be carried out.

A. SAMPLING

In reference to Step *I* of Fig. 2, the sample dataset $\boldsymbol{S}$, has been formulated from the complete custom-built numeric dataset of 100,000 records with 10 features as discussed above, having 67% of spam emails and the rest ham. The feature vector and the ratio of spam emails to ham in $\boldsymbol{S}$ is same as the original dataset $\boldsymbol{X}$. The purpose of sampling is to carry out the process of feature reduction through multiple unsupervised feature selection algorithms. Executing the feature selection algorithms on a dataset of 100,000 records with 10 features required a matrix manipulation of (100,000 × 10) data points which is rather infeasible and unscalable from hardware perspective required for such computation.

However, computation of a 50% sampling of $\boldsymbol{X}$ to $\boldsymbol{S}$, still requires considerable hardware capacity. To address that we deployed one of Amazon's AWS servers.

B. FEATURE REDUCTION THROUGH FEATURE ELIMINATION

With a large number of features, or attributes, model construction often becomes problematic due to several issues, such as Curse of Dimensionality [33], extended training time, overfitting etc. Feature Reduction\Selection tries to overcome these issues by logically selecting only those features which will have the most determining effect on the final output. Our feature vector, however, due to effective pre-processing and feature engineering, was already in a rather manageable state having 10 features only. However, we have sought to reduce it further by employing a novel ensemble feature reduction process- pictorially illustrated in Step *II* of Fig. 2.

The process deploys three Unsupervised Feature Selection algorithms; Principal Component Analysis (PCA), Laplacian Score for Feature Selection and Multi-Cluster-based Feature Selection (MCFS). There were few other options but we found those unscalable to reasonably large datasets. Before initiating the discussion on the proposed feature reduction

TABLE 1. A portion of the dataset.

Diff_FromDom ActDom	BL_IP	Created_within _1_year	Expire_in_ 13_months	mail_dt	cntry	msgid	tm	cn_typ	hop
1	0	0	0	0.9994024	0.9606935	0.5979188	0.7413460	0.8367861	2
1	0	0	0	0.9994024	0.9606935	0.6472665	0.9038769	0.8367861	2
1	1	0	0	0.0554284	0.8787605	0.3307662	0.9589951	0.7880419	3
0	0	0	0	0.0554284	0.9606935	0.4644148	0.9600159	0.4932314	2
1	0	0	0	0. 0554284	0.3464221	0.7190119	0.6930792	0.8367861	4
1	0	0	0	0. 0554284	0.3464221	0.9103321	0.3227595	0.8367861	4
1	1	1	0	0.7909815	0.6086028	0.0483820	0.2668493	0.8367861	2
0	1	0	0	0.1059830	0.8339643	0.0208947	0.5867826	0.8367861	1
0	1	1	0	0.2785842	0.6086028	0.3843499	0.5686232	0.8367861	1

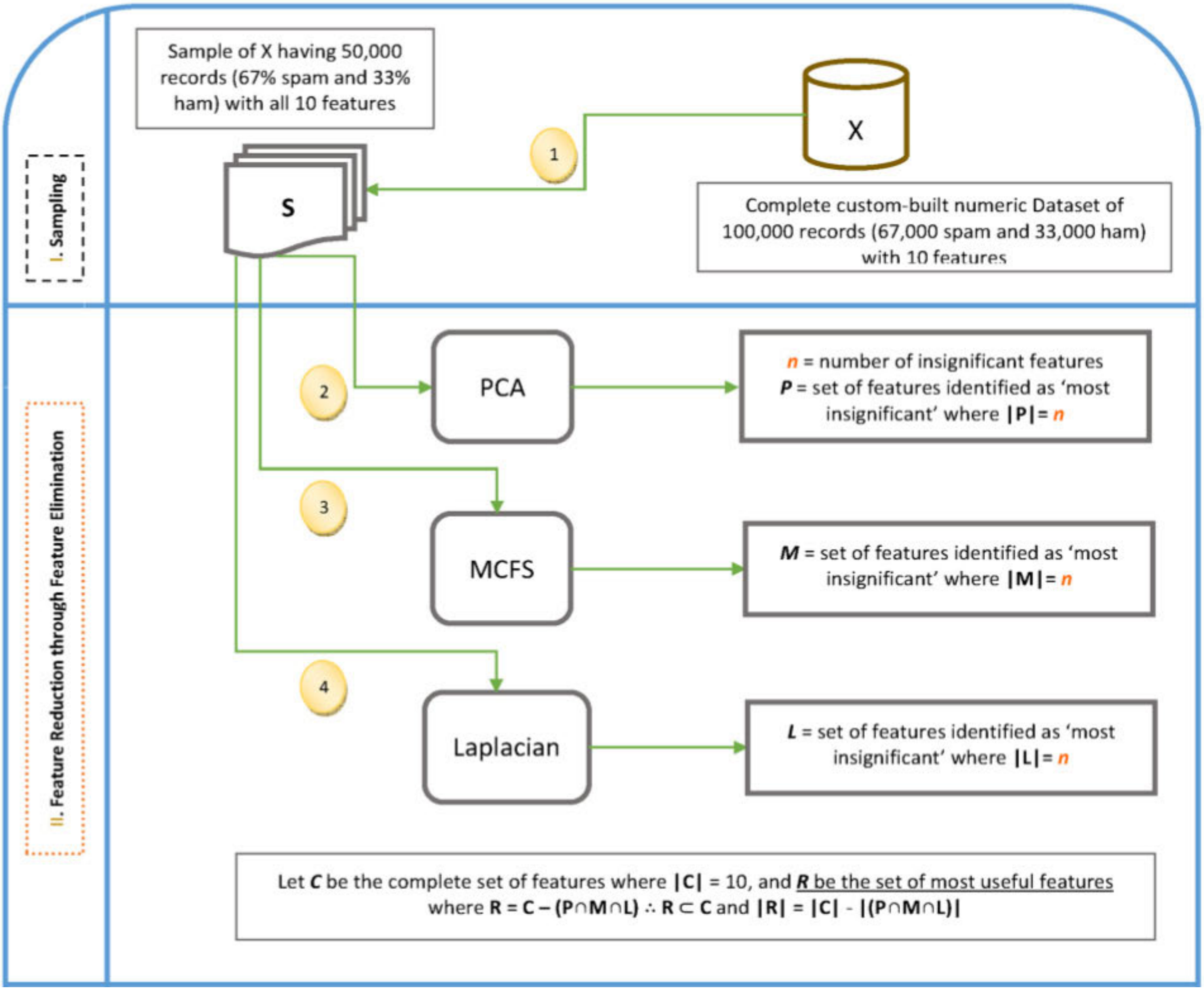

I. Sampling
Sample of X having 50,000 records (67% spam and 33% ham) with all 10 features
1
X
S
Complete custom-built numeric Dataset of 100,000 records (67,000 spam and 33,000 ham) with 10 features
II. Feature Reduction through Feature Elimination
2
PCA
n = number of insignificant features
P = set of features identified as 'most insignificant' where |P| = n
3
MCFS
M = set of features identified as 'most insignificant' where |M| = n
4
Laplacian
L = set of features identified as 'most insignificant' where |L| = n
Let C be the complete set of features where |C| = 10, and R be the set of most useful features where R = C – (P∩M∩L) ∴ R ⊂ C and |R| = |C| - |(P∩M∩L)|

FIGURE 2. A graphical breakdown of sampling and the proposed novel feature reduction techniques.

technique, we will have a brief and lucid discussion on the abovementioned three algorithms:

Principal Component Analysis (PCA): PCA is an unsupervised framework that works extremely well in most cases for 'Dimensionality Reduction' in such a fashion where the maximum variations of the dataset can be retained [35]. PCA is also a valuable tool in building Predictive Models. The system is an 'Orthogonal Linear Transformation' that transmutes the normalised inputted data to a new coordinate system [36]. To begin with, the labels are stripped off and the dataset is put into a Matrix $\boldsymbol{X}$. The 'Mean' - $\boldsymbol{X}$ is then calculated. Now the dataset is considered in its original form without stripping off the labels and 'Covariance Matrix' is calculated using (1) (showing Covariance of two variables P and Q that can be used to calculate Covariance of Matrix $\boldsymbol{X}$).

$$\mathbf{Cov(P, Q)} = \frac{\mathbf{1}}{\mathbf{n-1}}\sum_{\mathbf{i=1}}^{\mathbf{n}}(\mathbf{P_i}-\overline{\mathbf{P}})(\mathbf{Q_i}-\overline{\mathbf{Q}}) \tag{1}$$

The corresponding 'Eigenvalues' is then derived from the 'Eigenvectors', calculated from the Covariance Matrix of the original dataset and the value of $\boldsymbol{k}$ can be obtained by sorting the Eigenvectors by descending Eigenvalues, and taking $\boldsymbol{k}$ largest Eigenvectors where $\boldsymbol{k}$ is the number of dimensions of the newly obtained feature subspace and $\boldsymbol{k} <= \boldsymbol{d}$. Subsequently the projection matrix is fashioned from $\boldsymbol{k}$ Eigenvectors through which the original dataset $\boldsymbol{X}$ is transformed to obtain a new $\boldsymbol{k}$ -dimensional feature subspace. In this research, PCA has been used before MCFS and Laplacian to get the cardinality of the feature-set holding the least impactful features.

Multi-Cluster-Based Feature Selection (MCFS): MCFS selects a subset of the original feature-set based on the optimisation over an '$L1-$regularized least-squares' problem [37]. A key aspect of the algorithm is its ability to maintain the multi-cluster structure of the data. Determining the correlations between different features is carried out by spectral analysis without any corresponding labels. The spectral analysis usually clusters the data points using the top eigenvectors of *graph Laplacian* (*discussed more in the 'Spectral Clustering' section*). MCFS calculates the linear reflection of low-dimension representation of high-dimension features by resolving the $L1-$ regularized regression problem [37] as shown in (2).

$$\min_{a_k} ||p_k - X^T a_k||^2 + \beta |a_k| \tag{2}$$

In (2), $\boldsymbol{P}$ is the 'flat' embedding for the data points where $\boldsymbol{P} = [p_1, p_2, \ldots, p_k]$, a_k is the $\boldsymbol{N}$-dimensional vector and $|a_k| = \sum_{i=0}^{N} |a_{ki}|$ denotes the $L1$-norm of a_k.

Eventually the most useful features are selected having the maximum coefficient of sparse representation and assigning a corresponding score called MCFS score. For every feature $\boldsymbol{m}$, the corresponding MCFS score, $\boldsymbol{C}$, is attributed using (3), where $a_{k,m}$ is the m^{th} member of vector a_k.

$$\mathrm{C}(m) = \max_{a_k} |a_{k,m}| \tag{3}$$

All the features are then sorted in descending order on the basis of $\boldsymbol{C}$. This algorithm is quite useful while the number of features is less than fifty [38].

Laplacian Score For Feature Selection: The algorithm works on the belief that data residing in the same class are often close to each other; thus importance of a feature can be determined by its power of locality preservation. The algorithm starts off by embedding the data on a nearest neighbor graph $\boldsymbol{G}$ having $\boldsymbol{n}$ nodes. The $\boldsymbol{i}^{\text{th}}$ node represents the element $\boldsymbol{x_i}$. The graph makes a connection to $\boldsymbol{x_i}$ with another element or node $\boldsymbol{x_j}$, belonging to k nearest neighbors of $\boldsymbol{x_i}$. The Weight Matrix, $\boldsymbol{W}$ of $\boldsymbol{G}$ describes the local structure of the data space and is defined using (4).

$$W = \begin{cases} e^{-\frac{||x_i - x_j||^2}{c}}, & \textit{if } x_i \in kNN\left(x_j\right) \textit{ or} \\ & \textit{vice versa} \\ \mathbf{0} & \textit{otherwise} \end{cases} \tag{4}$$

$\boldsymbol{c}$ is an appropriately chosen constant, and a *graph Laplacian*, $\boldsymbol{L}$, is constructed from $\boldsymbol{W}$ (*graph Laplacians are discussed in the 'Spectral Clustering' section*). A Laplacian score, $\boldsymbol{LS}$, is then calculated for each feature using (5) and ranked accordingly [39]. Equation (5) demonstrates the Laplacian Score, $\boldsymbol{LS}$, of a feature $\boldsymbol{k}$, where $\boldsymbol{D}$ is the *Diagonal Matrix* of $\boldsymbol{W}$

$$LS_k = \frac{(f_k - \frac{f_k^T D1}{1^T D1} * 1)^T * L * (f_k - \frac{f_k^T D1}{1^T D1} * 1)}{(f_k - \frac{f_k^T D1}{1^T D1} * 1)^T * D * (f_k - \frac{f_k^T D1}{1^T D1} * 1)} \tag{5}$$

Pseudocode 1 Pseudocode for Feature Reduction Algorithm

```
BEGIN
  1. LET {P}, {M}, {L}, {C}, {R} = { }
  2. PCA (S)
       a. {P} = Set of least important features across the
          most important Principal Components
  3. LET $n$ = Cardinality of P or |P|
  4. MCFS (S)
       a. {M} = Set of $n$ number of least important features
  5. Laplacian_Score (S)
       a. {L} = Set of $n$ number of least important features
  6. {C} = Set of all 10 features in S
  7. {R} = C – (P ∩M ∩L)
END
```

1) PROPOSED METHOD FOR FEATURE REDCUTION

As has been mentioned before we will be using PCA first to get the set of least important features spread within the number of principal components that will represent the majority of the sample dataset $\boldsymbol{S}$, in addition, the cardinality of that set, $\boldsymbol{n}$ is also important as for MCFS and Laplacian Score, we will take $\boldsymbol{n}$ number of least important features to formulate the sets corresponding to those two algorithms.

From Pseudocode 1 it is evident that three sets of least impactful features have been identified using the three algorithms and the set $\boldsymbol{R}$ (containing the most useful features for clustering purposes) has been derived such that it holds all the features from the original feature-set of 10 excluding those found in common within the three sets $\boldsymbol{P}$, $\boldsymbol{M}$ and $\boldsymbol{L}$ of least

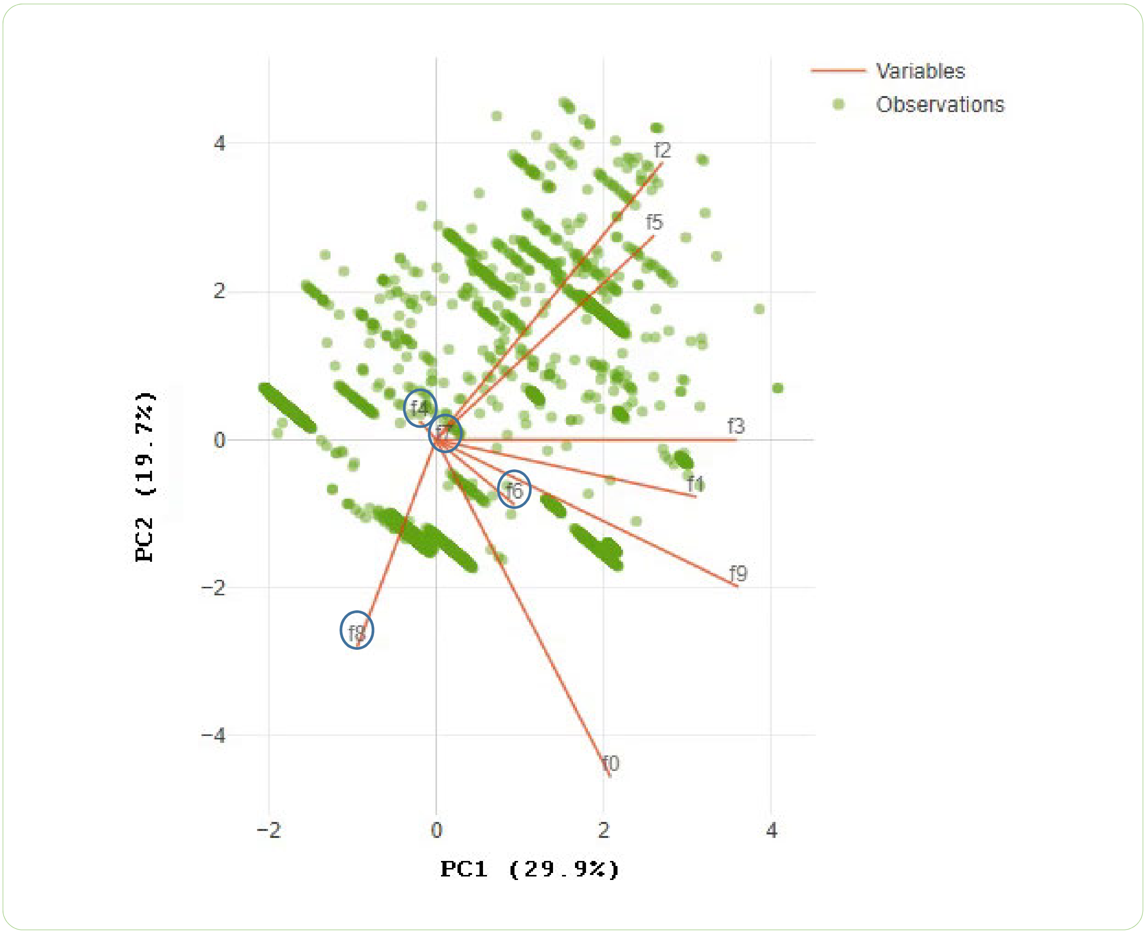


FIGURE 3. Biplot of pca applied on the sample dataset.

important features. In addition, Cardinality of {**R**}, |**R**| = |**C**| - |(**P**∩**M**∩**L**)|.

2) RESULT EVALUATION OF FEATURE SELECTION ALGORITHMS

In this section we will detail out the results obtained after each of the algorithms has been applied to dataset $\boldsymbol{S}$.

a: PRINCIPAL COMPONENT ANALYSIS (PCA)

In almost universally, in case of PCA, the first two principal components (PC1 and PC2) represent or explain majority of the data (over 50%). Because of our ensemble based approach for feature selection, we designed the approach giving *as much flexibility as possible* in using PCA, as this flexibility or room is required for the functioning of the other two algorithms. Eventually, the least important features will anyway be identified with confidence and will be left out. Our approach has been to use the first two Principal Components as long as those can cumulatively explain at least 40% of the data. In case it falls below 40% (in rare instances), we will drop PCA altogether and will only use MSFC and Laplacian Feature Selection to look for consensus on the least *two* significant features; that is the value of $\boldsymbol{n}$ will be 2 to begin with.

Now provided that at least 40% of the data can be represented by PC1 and PC2, those feature(s) having cumulative 'weight' across PC1 and PC2 falling below a preset threshold value of 15% (or 0.15), will be identified as 'least important' (PCA only). This threshold value of 15% has been established keeping in mind that there should be a 'degree of freedom'- as we may get tempted to choose a value that is too low (for instance < 10%). This will render the recommendations of MCFS and Laplacian Score rather ineffective. The cardinality of the three sets in that case may become rather tight, and after commonality comparison, the process may leave out feature(s) which is/are in fact 'not effective' from the final set of *least* important feature(s). Our aim is to eliminate those low-ranked feature(s) which are deemed non-essential across multiple feature selection algorithms; thereby giving a high degree of confidence in $\boldsymbol{R}$, the set of most useful features. Obviously, for this hypothesis to be acceptable, we will have to evaluate how the clustering algorithms used in this research respond to $\boldsymbol{R}$.

Once the PCA has been applied, it can be observed from Fig. 4 that the first two principal components (PC1 and PC2) account for nearly 50% of the data. The Biplot (Fig. 3) also suggests that f_4, f_6 and f_7 have rather a minor variability across PC1 and PC2, and carry insignificant 'Weight'.

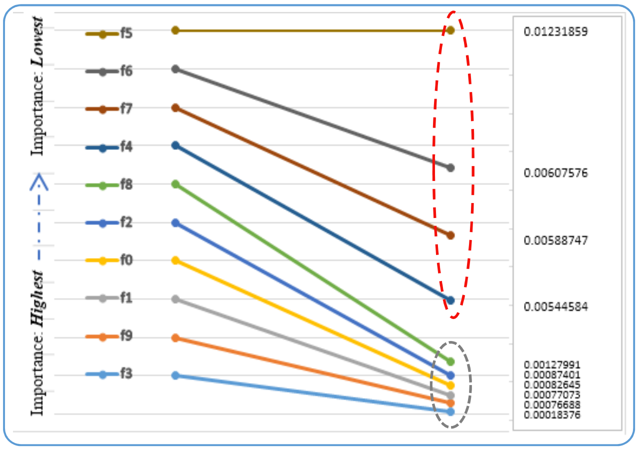


FIGURE 4. A line plot of laplacian scores for each feature.

TABLE 2. Weights for each of the features on PC1 and PC2.

FEATURE	PC1	PC2	PC1 + PC2
f_0	0.08	0.38	0.46
f_2	0.13	0.25	0.38
f_9	0.23	0.07	0.30
f_5	0.12	0.14	0.26
f_3	0.23	0.01	0.24
f_1	0.17	0.01	0.18
f_8	0.00	0.14	0.14
f_6	0.02	0.01	0.03
f_4	0.00	0.01	0.01
f_7	0.00	0.00	0.00

Table 2 projects the PCA-produced ‘Weight Matrix’ for all 10 features. It can clearly be observed that f_4, f_6 and f_7 indeed have very low cumulative weights (0.01, 0.03 and 0.00). Besides, f_8 also seem to have a cumulative weightage, (**PC1** + **PC2**), of only 0.14, thus failing to push through the threshold value of 0.15. We therefore consider these four features to be included in the set of least impactful features, $\boldsymbol{P}$, obtained from PCA. Therefore, $\boldsymbol{P} = \{f_6, f_4, f_7, f_8\}$ and $\boldsymbol{n} = |\mathbf{P}|$ or 4.

Thus for Laplacian Score and MCFS, the last ranked n number of features or four (4) features will be selected as the elements for the sets of least important features respectively.

b: LAPLACIAN FEATURE SCORE

Table 2 lists the features from most important (f_3) to the least important (f_5) according to the ranking based upon the scores assigned to each feature.

As we can see from Table 3, the lowest ranking four (n) features are f_4, f_7, f_6, and f_5. Therefore in the set L of least impactful features for Laplacian Feature Score, the elements will be $\boldsymbol{L} = \{f_4, f_7, f_6, f_5\}$.

TABLE 3. Scores for each of the features on Laplacian score.

FEATURE	Score
f_3	0.000183
f_9	0.000766
f_1	0.000770
f_0	0.000826
f_2	0.000874
f_8	0.001279
f_4	0.005445
f_7	0.005887
f_6	0.006075
f_5	0.012318

TABLE 4. Scores for each of the features on MCFS score.

FEATURE	Score
f_8	0.013954
f_3	0.006262
f_7	0.003179
f_0	0.002726
f_1	0.001551
f_9	0.001412
f_5	0.001316
f_6	0.000590
f_2	0.000481
f_4	0.000269

Another interesting aspect for this instance can be observed by plotting a simple line plot of Laplacian Scores (Fig. 4) where the more impactful features are almost equally contributing as those are tightly fitted in a cluster; while the less important features are rather distant and consequently the levels of impact are quite minimal in comparison to the cluster of useful features.

c: MULTI-CLUSTER FEATURE SELECTION (MCFS)

Table 4 lists the features from most important (f_8) to the least important (f_4) according to the corresponding MCFS scores.

As we can see from Table 4, the lowest ranking four (n) features are f_5, f_6, f_2, and f_4. Therefore in the set L of least impactful features for MCFS, the elements will be $\boldsymbol{M} = \{f_5, f_6, f_2, f_4\}$.

d: GENERATING THE LIST OF MOST USEFUL FEATURE

After the successful executions of all three feature selection algorithms, we have the following sets of least useful features:

$$\begin{aligned} \text{PCA}, \boldsymbol{P} &= \{f_4, f_6, f_7, f_8\} \\ \text{Laplacian}, \boldsymbol{L} &= \{f_4, f_7, f_6, f_5\} \\ \text{MCFS}, \boldsymbol{M} &= \{f_5, f_6, f_2, f_4\} \end{aligned}$$

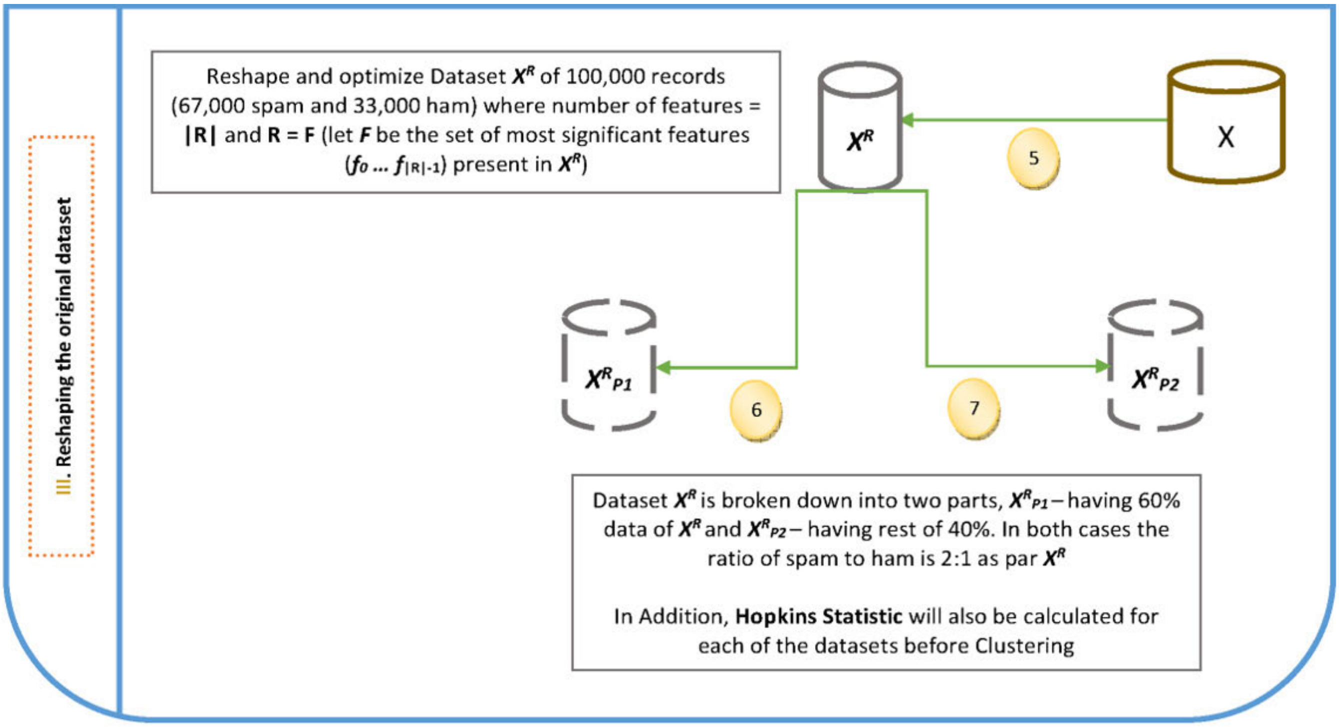

III. Reshaping the original dataset
Reshape and optimize Dataset X^R of 100,000 records (67,000 spam and 33,000 ham) where number of features = |R| and R = F (let F be the set of most significant features ($f_0 \dots f_{|R|-1}$) present in X^R)
X^R
5
X
X^R_{P1}
6
7
X^R_{P2}
Dataset X^R is broken down into two parts, X^R_{P1} – having 60% data of X^R and X^R_{P2} – having rest of 40%. In both cases the ratio of spam to ham is 2:1 as par X^R
In Addition, **Hopkins Statistic** will also be calculated for each of the datasets before Clustering

FIGURE 5. Reshaping the original dataset to create X^r_{P1} and X^r_{P2}.

If we consider C to be the set of all ten (10) features, then the set of most useful features, R, is derived as per (6):

$$\{R\} = \mathbf{C} - (\mathbf{P} \cap \mathbf{M} \cap \mathbf{L})$$
$$\therefore R = \{f_0, f_1, f_2, f_3, f_5, f_7, f_8, f_9\} \tag{6}$$

Thus features f_4 and f_6 have been identified by all three algorithms as less impactful features, and we can confidently express that the set of $\mathbf{R}$ indeed contains smallest feature subset and all the elements (features) in $\mathbf{R}$ is carrying significant degree of contributing weights that can aid in revealing useful clusters. The cardinality of $\mathbf{R}$ or $|\mathbf{R}| = 8$, so we have managed to achieve a 20% reduction from the original feature vector ($\mathbf{C}$) of cardinality 10, retaining 80% of the most useful features. f_4 and f_6 in the original feature vector stand for 'date of email' and 'Message-ID' respectively. We will now have to restructure our original dataset $\mathbf{X}$, leaving these two less important features out.

C. RESHAPING THE ORIGINAL DATASET

The bonafide dataset, X, as shown in Fig. 5, at this stage has been transformed into $\mathbf{X^r}$, having the feature-set of $\mathbf{R}$. So instead of the 10 features, it now has 8 of the most critical ones. Afterwards, $\mathbf{X^r}$ has been sliced into two separate datasets, $\mathbf{X^r_{P1}}$ and $\mathbf{X^r_{P2}}$. $\mathbf{X^r_{P1}}$ contains 60% or 60,000 data rows inclusive of ham and spam emails (1:2) and $\mathbf{X^r_{P2}}$ houses rest of the 40% data with the same ratio.

Both have the same feature vector $\mathbf{R}$. The reason we have decided to separate the datasets as once the clustering algorithms have clustered dataset $\mathbf{X^r_{P1}}$, and results are being validated through rigorous measures, it is important to evaluate the degree the consistency of this validated clustering results across another different set of data. If there is significant consistency projected by any of the algorithms, we can reach a decision on the performance of the algorithm with a high degree of confidence.

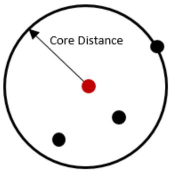


FIGURE 6. Core distance.

D. APPLYING UNSUPERVISED CLUSTERING ALGORITHMS TO CREATE THE CLUSTERS

This section will briefly discuss the algorithms that have been used for clustering purposes (Fig. 7). As mentioned before, only those algorithms where the number of clusters created can be controlled, have been deployed for the model. In the subsequent sub-sections, the resulting clusters will be investigated through 3D Scatterplot visualisation.

1) CLUSTERING ALGORITHMS USED

a: BIRCH (BALANCED ITERATIVE REDUCING AND CLUSTERING USING HIERARCHIES)

BIRCH is one of those very few unsupervised clustering algorithms that can cluster substantially large datasets with available (and often limited) resources such as 'main memory'

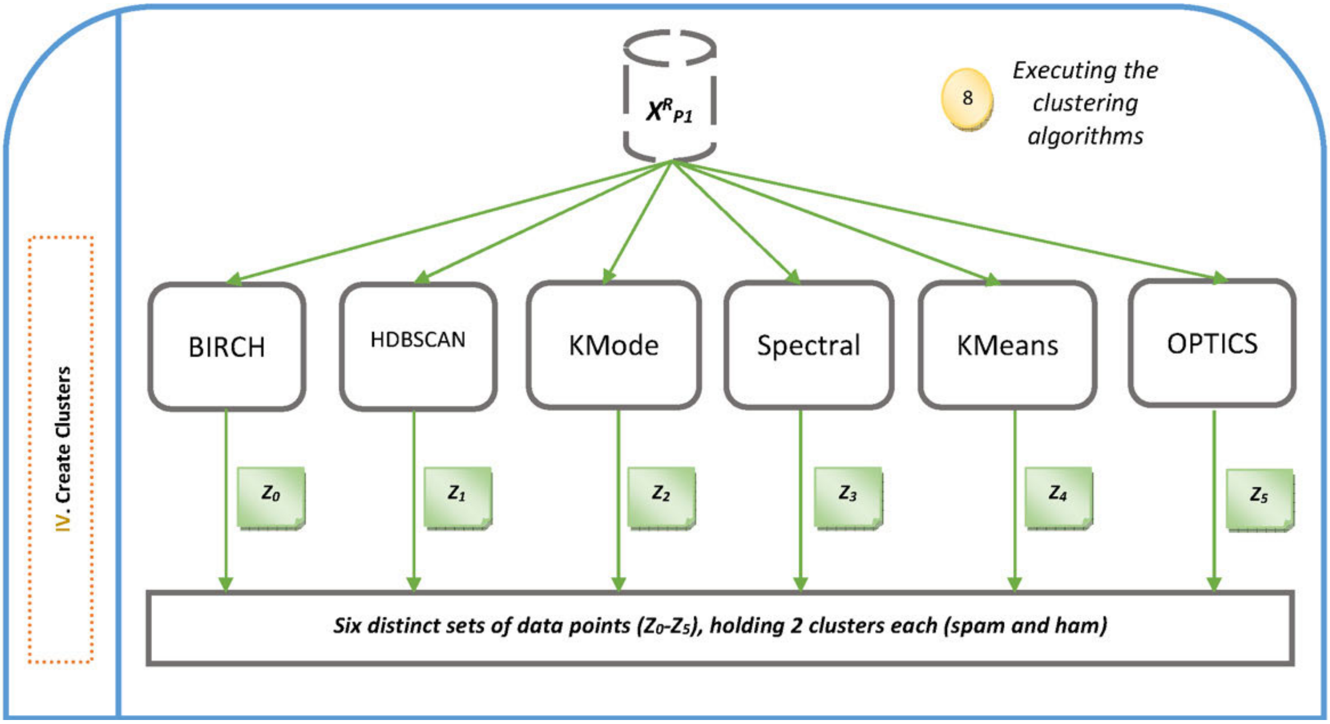

IV. Create Clusters
X^R_{P1}
8
Executing the clustering algorithms
BIRCH
HDBSCAN
KMode
Spectral
KMeans
OPTICS
Z_0
Z_1
Z_2
Z_3
Z_4
Z_5
Six distinct sets of data points (Z_0-Z_5), holding 2 clusters each (spam and ham)

FIGURE 7. The application of clustering algorithms ion X^r_{P1}.

of a processing unit; In general, BIRCH incrementally and dynamically processes an un-clustered dataset of multi-dimensional metric data points in a rather time-efficient manner in one single scan [40]. The clustering quality is often thereafter improved through some additional scans. The algorithms is said to effectively handle data points that are not really part of the underlying patterns (often denoted as 'Noise'). To handle large datasets, BIRCH first calculates a 'Triple' entries for data points known as 'Clustering Feature (CF)', then dynamically building a tree of CFs (CF-Tree). Given $\boldsymbol{N}$ $\boldsymbol{d}$-dimensional data points, emails in this case, in a cluster $\{\overrightarrow{X}\}_j$, where $j = 1, 2, \ldots, N$, the CF entry of the corresponding cluster is determined as the 'Triple' $\{\boldsymbol{N}, \overrightarrow{\boldsymbol{LS}}, \boldsymbol{SS}$, where $\boldsymbol{N}$ is the total number of data points in the cluster, $\boldsymbol{SS}$ represents the square sum of $\boldsymbol{N}$ data points, i.e. $\sum_{j=1}^{N} \overrightarrow{X}_j^{2}$, and $\overrightarrow{\boldsymbol{LS}}$ is the linear sum of $\boldsymbol{N}$ data points, i.e. $\sum_{j=1}^{N} \overrightarrow{X}_j$ [40]. A CF may also be composed of multiple other CFs. CF-Tree is a compact summary of the complete dataset and retains maximum distribution information of the data. Subsequent incremental clustering is then carried out on this summary representation instead of the original dataset [40].

b: HDBSCAN (HIERARCHICAL DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE)

An extension and improved version of DBSCAN algorithm where 'Density based Clustering' technique has been preferred over centroid-based clustering as in K-means. Density based clustering is an unsupervised learning technique that recognises distinctive clusters in the data, on the assumption that a cluster in a data space is a contiguous region of high point density, alienated from other such dense areas of clusters by contiguous areas of low point density [41].

HDBSCAN first gets a rough estimate of 'density' and then 'pushes away' the points in low dense areas further from not only each other, but also regions of high density. 'Mutual Reachability Distance (MRD)', shown in (7) is used to achieve the purpose.

$$x_{\text{mrd-k}}(p, q) = \max\{\text{core}_k(p), \text{core}_k(q), x(p, q)\} \tag{7}$$

core_k(p) or the 'Core Distance' is measured for parameter **k** for a point **p** as distance that is required to travel from each point to the defined minimum number of points for a cluster. Fig. 6 shows the concept of Core Distance for $\mathbf{k} = 4$.

So, if a 'large' minimum points per cluster is selected, then the corresponding 'core distance' will also be larger. $x(p, q)$ is the original metric distance between **p** and **q**. Now the dense points having low core distance will remain the same distance apart from each other but sparser points will be pushed apart to be at a minimum their core distance away from any other point. The algorithm then employs a Minimum Spanning Tree [42] to identify the dense regions, builds up a hierarchy of clusters and condenses those as required before extracting the final clusters. HDBSCAN works well even when clusters are arbitrarily shaped and of dissimilar density and sizes.

c: K-MEANS

One of the most used and common centroid-based clustering algorithms around that attempts to cluster similar data points together to find any underlying pattern. K-means delivers the final output through a process called iterative refinement. It tries to minimise the sum of the squared distance between the data points and the cluster's centroid. 'Centroid' is defined as the arithmetic mean of all the data points that belong to that cluster. The number of groups is denoted by K, and iteratively each data point is assigned to one of these groups of clusters based on the identified similarities among the features [43]. The Initial number of clusters 'K' has to be provided as an input. It can sometimes be a delicate issue and users often end up running the system multiple times with different values of K, and afterwards a comparison is drawn to select the best value of 'K'. However, various methods are available for getting a reasonably stable approximation of K [43]. K-means most commonly uses 'Euclidean Distance' to determine the distance between two data points ($\boldsymbol{Z^n}$ and $\boldsymbol{Z^m}$) as shown in (8) [44]. One of the key advantages of K-means is that in case number of features are really high, it can still complete the computation in a reasonable time if the value of 'K' is kept rather small.

$$Dist\left(Z^n, Z^m\right) = \sqrt{\sum_{i=1}^{D} (Z_i^n - Z_i^m)^2} \tag{8}$$

Given a set of d-dimensional real vector observations (y_1, $y_2, \ldots, y_n$), K-means clustering targets, as shown in (8), at partitioning the $\boldsymbol{n}$ observations into $\boldsymbol{k}$ ($\leq \boldsymbol{n}$) sets $\boldsymbol{S} = \{\boldsymbol{S_1}, \boldsymbol{S_2}, \ldots, \boldsymbol{S_k}\}$ so as to minimise the Variance. $\boldsymbol{\mu_i}$ denotes the 'Mean' of S_i and $\boldsymbol{V}$ is the Variance in (9).

$$\underset{S}{arg\,min} \sum\nolimits_{i=1}^{k} \sum\nolimits_{y\in S_1} ||y - \mu_i||^2 = \underset{S}{arg\,min} \sum\nolimits_{i=1}^{k} |S_i|\, V\, (S_i) \tag{9}$$

d: SPECTRAL

Spectral clustering is gaining considerable grounds in recent years due to its straightforward implementation and encouraging performance especially in graph-based clustering, which quite often outperforms other frequently deployed algorithms such as the K-means.

Spectral clustering starts off by generating a 'Similarity graph' between inputted **N** objects to cluster. Then it defines a feature vector for each of the **N** object by computing the first **k** eigenvectors of its Laplacian matrix before executing a K-means on these features to group objects into **k** clusters. Epsilon neighbourhood graphs, or ε- neighbourhood graphs is the most common method of building the 'Similarity graph' which is a non-negative symmetric graph. Each vertex is connected to vertices falling inside a real-valued circular radius ε (epsilon), which requires necessary tuning to capture the local structure of data.

The crux of the algorithm are the graph Laplacian matrices [45], **L**, as demonstrated in (10).

$$L = D - A \tag{10}$$

where, **A** is a 'Adjacency matrix' having $\mathbf{A_{ij}} \geq 0$ of graph **G**. **D** is the 'Diagonal matrix' of **A**. A normalised form of Laplacian matrix, $\mathbf{L_{pq}}$, is often defined as in (11), where **d** are the points in the Diagonal matrix.

$$L_{pq}(G) = \begin{cases} 1 & \text{if } i = p \text{ and } q \neq 0 \\ -\dfrac{1}{\sqrt{d_p d_q}} & \text{if } p \text{ and } q \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

e: K-MODES

Unlike K-means, K-modes applies a straightforward measure of matching dissimilarity for categorical data. Additionally, instead of using 'means' for centroid creation, K-modes relies on 'Mode' statistics. The algorithm calls upon frequency-related strategy on these 'modes' to limit the clustering costs as much as possible and K-modes banks on frequency-related techniques to minimise the costs. These differences from K-means accounts for its ability to handle pure unconverted categorical data. [46].

A dissimilarity measure for $\mathbf{Y^1}$ and $\mathbf{Y^2}$, two **n**-dimensional vectors, can be obtained through (12). A higher number of mismatches will clearly indicate the lower degree of similarity between $\mathbf{Y^1}$ and $\mathbf{Y^2}$. This *dissimilarity* **d**, can be expressed as:

$$\boldsymbol{d}\left(\boldsymbol{Y^1}, \boldsymbol{Y^2}\right) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \delta(y_j^1, y_j^2) \tag{12}$$

with

$$\delta(y_j^1, y_j^2) = \begin{cases} \mathbf{0}, & \textbf{if } y_j^1 = y_j^2 \\ \mathbf{1}, & \textbf{if } y_j^1 \neq y_j^2 \end{cases}$$

f: OPTICS (ORDERING POINTS TO IDENTIFY CLUSTER STRUCTURE)

Another useful Density-based clustering algorithm that has a close similarity to HDBSCAN in the way it works. OPTICS also works well with varying cluster sizes. Core samples of high density are first identified and subsequently expanded into multiple clusters from those core samples. An ordering of all objects in a given dataset is initially calculated to begin with. Now for each of the objects or points in that dataset, core-distance and an appropriate 'Reachability distance' is stored. Reachability distance, $\boldsymbol{r\_d}$, of an object or point $\boldsymbol{y}$ with respect to another object or point $\boldsymbol{x}$ is the smallest distance from $\boldsymbol{x}$ if $\boldsymbol{x}$ is a *core object* (objects having dense neighbourhood). Basically $\boldsymbol{x}$ is a core object or point if at least $\boldsymbol{min\_pts}$ points are found, including itself, within its ε-*neighbourhood* $\boldsymbol{N_\epsilon}$ $(\boldsymbol{x})$. It also cannot be smaller than the core distance, $\boldsymbol{c\_d}$, of **y** as demonstrated in (13) [47]. Epsilon, ε, denotes the maximum distance to consider, while $\boldsymbol{min\_pts}$ indicates the minimum number of points needed to form a cluster.

$$\boldsymbol{r\_d}_{\boldsymbol{\varepsilon,min\_pts}}(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} \boldsymbol{undefined} & \\ \boldsymbol{max}(\boldsymbol{c\_d}_{\boldsymbol{\varepsilon,min\_pts}}(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y})) & \begin{cases} \boldsymbol{if}\ |\boldsymbol{N_\epsilon}\ (\boldsymbol{x})| \\ < \boldsymbol{min\_pts} \\ \boldsymbol{otherwise} \end{cases} \end{cases} \tag{13}$$

TABLE 5. Hopkins statistic for the datasets.

DATASET	Hopkins Probability	Clustering Tendency
X^r_{P1}	≈ 91.4%	Strong for both the cases, that is, probability of having meaningful clusters are high
X^r_{P2}	≈ 92.0%	

OPTICS then maintains a list known as OrderSeeds to produce the output ordering. Objects in OrderSeeds are sorted by the reachability-distance from their respective closest core objects. OrderSeeds is a linear list of all objects under analysis and represents the density-based clustering structure of the data, from which basic clustering information of that dense area, such as shape and centroid can be retrieved.

2) DETERMINING CLUSTERING TENDENCY OF THE DATASETS

Before applying any clustering algorithm to our datasets $\mathbf{X^r_{P1}}$ and $\mathbf{X^r_{P2}}$, evaluation using Hopkins Statistic [48] had been carried out just to confirm the presence of non-random, relevant cluster-like structures within the data. That is, whether our datasets indeed have *meaningful* clusters to begin with. Table 5 summarises the findings indicating high confidence in the presence of relevant clusters as the probabilities are above 90%. Hopkins statistic is basically a spatial measure that tests the spatial randomness of a variable as distributed in a space [58].

3) CLUSTERS PRODUCED BY THE ALGORITHMS

In a perfect world, if an algorithm can deliver 100% accurate results, would produce the clusters as shown in Fig. 8.

As can be seen in Fig. 8, all the data points from $\mathbf{X^r_{P1}}$ form perfect clusters where no overlapping can be detected. That is, no records (data points) of spam emails have been misclassified into ham, or in the context of the 3D perspective of the figure, we can say 'data points not going up from bottom plane to z_1 area (marked in red dotted box)', and no ham have also been misclassified as spam emails, or in the context of the 3D perspective of the figure, we can say 'data points not going down to the to the bottom plane at z_0 area region'. However, in real world it would not be as perfect as this as there are bound to be some misclassfications, and subsequent admixture of data points from both clusters. In the following section, we will visually examine how the algorithms clustered $\mathbf{X^r_{P1}}$. The algorithms that get closer to the above perfect-clustering, implies better performance.

a: CLUSTERS PRODUCED BY BIRCH

Fig. 9(a) and Fig. 9(b) display the two clusters generated by BIRCH from top-down and bottom-up view. The images tell us that the clustering is actually quite poor as a large number of ham have been misclassified as spam whereas some degree of spam emails have also been misclassified as ham. The overall clustering achieved by BIRCH thus remains far from perfect in this case. The high degree of ham getting misclassified is the key issue here that clearly caused the quality of overall clustering to plummet. The red dotted line along with the yellow signage in the following scatterplots approximately indicates the area of misclassification.

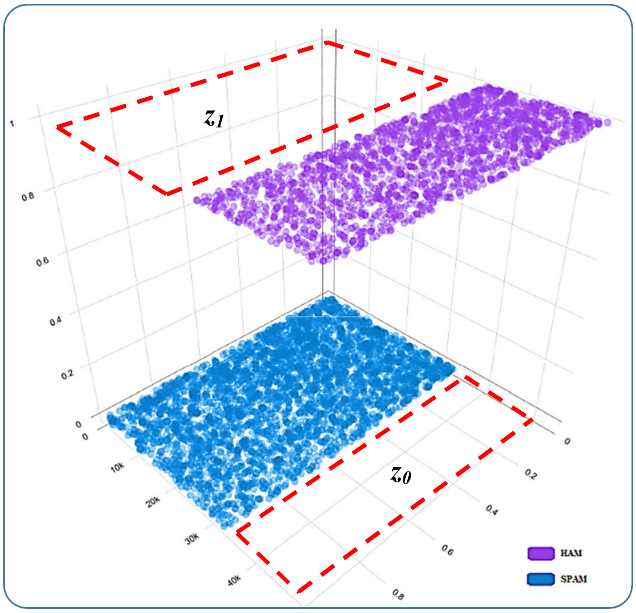


FIGURE 8. Perfect clusters in an ideal scenario.

b: CLUSTERS PRODUCED BY HDBSCAN

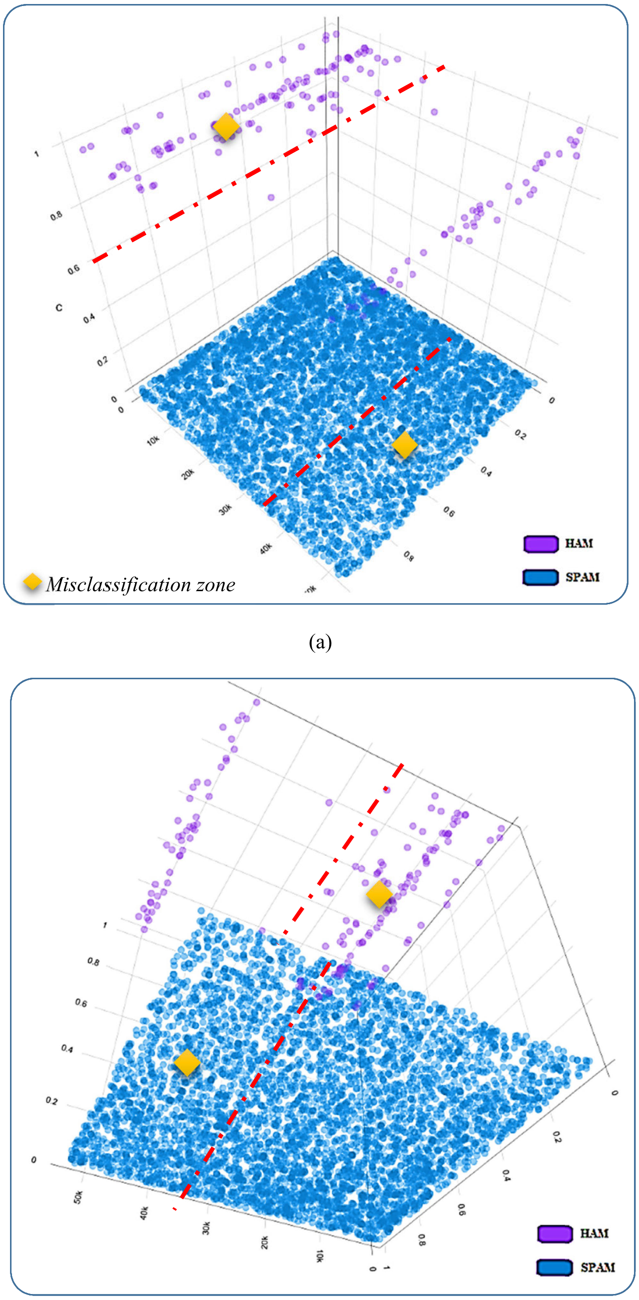
Clusters generated by HDBSCAN did not show strong performance either, as can be seen from Fig. 10 (a) and 10 (b) In this case it has been the other way around. High degree of spam emails had been misclassified as ham and some degree of ham into spam emails. As HDBSCAN does not accept the exact number of clusters as a parameter, we have set the 'min_cluster_size' as the 17% of the inputted data, leading to a 2-cluster solution.

c: CLUSTERS PRODUCED BY K-MODES

K-modes performs better in the context of BIRCH and HDBSCAN as projected in Fig. 11 (a) and 11 (b), but still considerable overlapping can be seen in both the regions of misclassifications, thus both Ham and spam emails have been wrongly identified to quite a large extent which indicates K-modes' performance is still not up to the scratch and thus keeps the door open for other clustering algorithms.

d: CLUSTERS PRODUCED BY SPECTRAL

Spectral clustering has been visualised in Fig. 12 (a) and Fig. 12 (b). Corresponding figures show that Spectral seems to have achieved significantly better clustering in comparison to the algorithms we have investigated thus far. Regions of



(a)

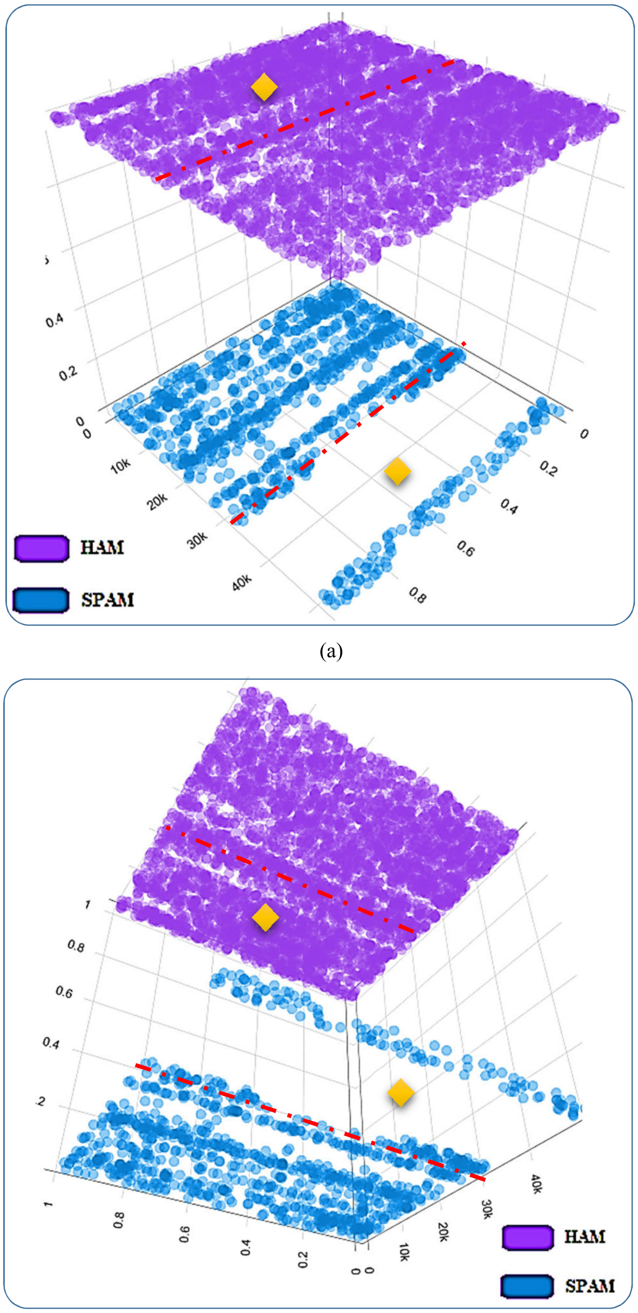
(b)

FIGURE 9. (a). Top-down view of clusters (BIRCH). (b). Bottom-up view of clusters (BIRCH).

misclassifications are mostly empty except few strips consisting of misclassified data points. Overall it seems to have performed better than K-modes, and most certainly outperformed both BIRCH and HDBSCAN.

e: CLUSTERS PRODUCED BY K-MEANS

The performance demonstrated by K-means as can be seen in Fig. 13 (a) and 13 (b) is comparable to Spectral. The clustering structures in both the cases is almost similar. Thus when we go to validation of results, we can then quantify the performance of these two algorithms and deduce which one performed better over others.



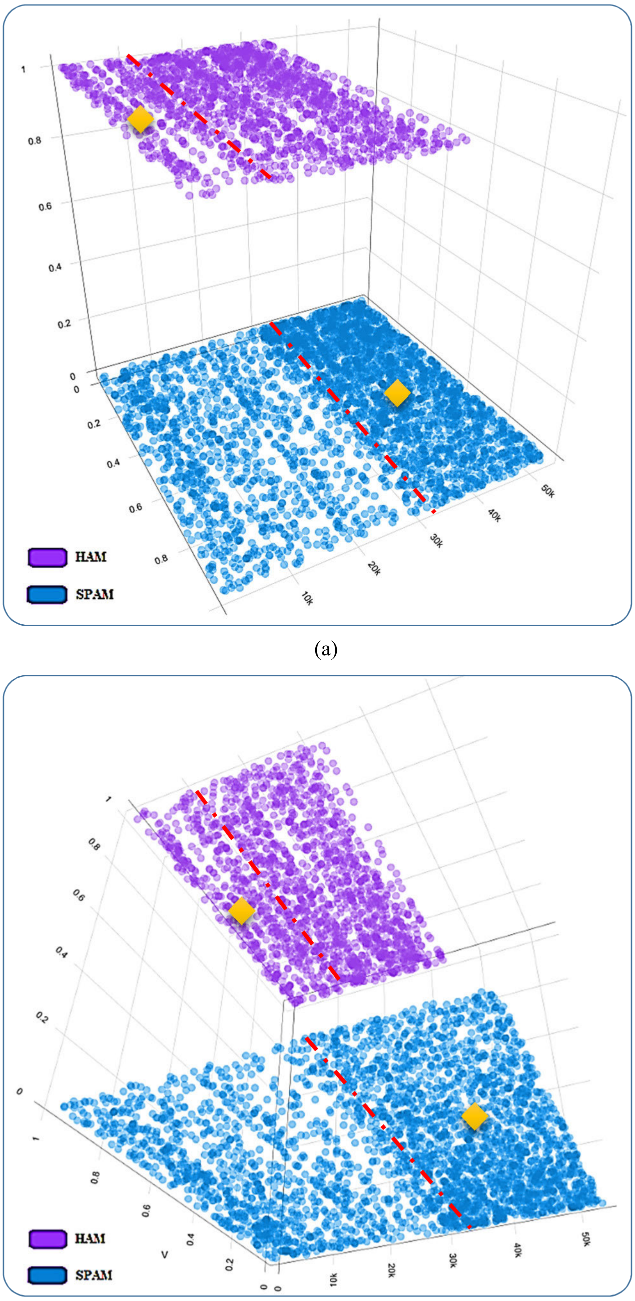
(a)

(b)

FIGURE 10. (a). Top-down view of clusters (HDBSCAN). (b). Bottom-up view of clusters (HDBSCAN).

f: CLUSTERS PRODUCED BY OPTICS

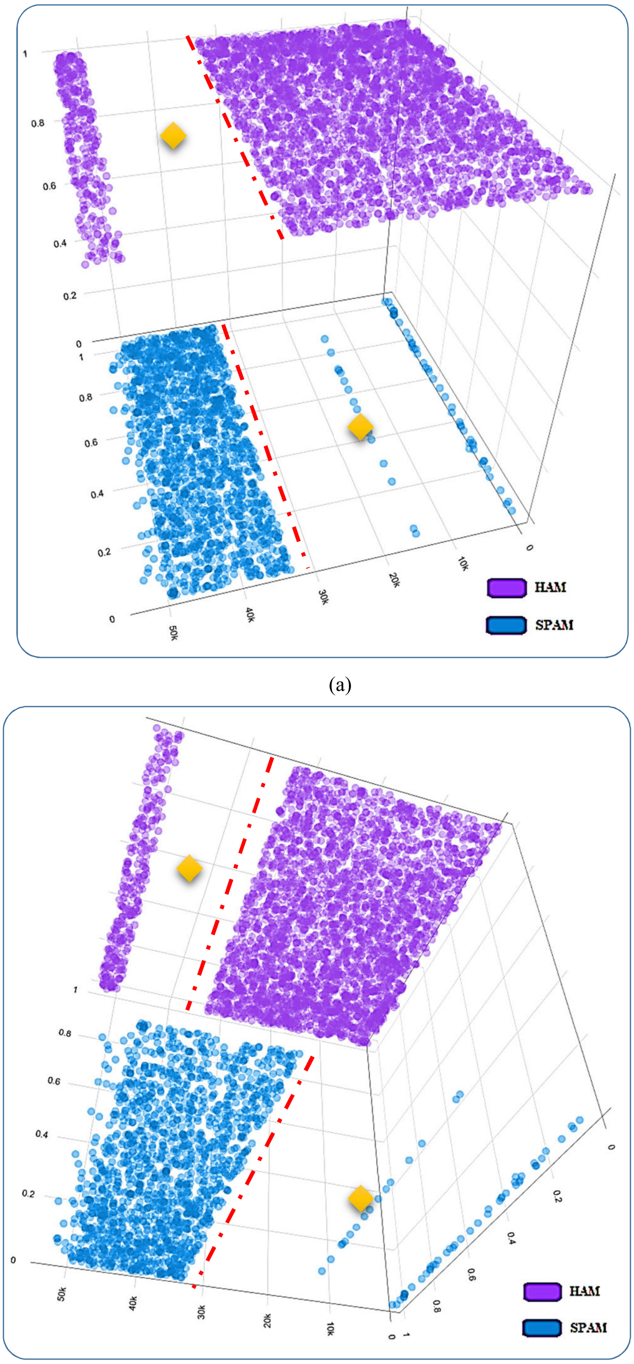
OPTICS, from Fig. 14 (a) and 14 (b), seems to have surpassed all the other algorithms analysed thus far and as it appears to be from visualisation, produced the most compact set of clusters.



(a)

(b)

FIGURE 11. (a). Top-down view of clusters (K-MODES). (b). Bottom-up view of clusters (K-MODES).



(a)

(b)

FIGURE 12. (a). Top-down view of clusters (spectral). (b). Bottom-up view of clusters (spectral).

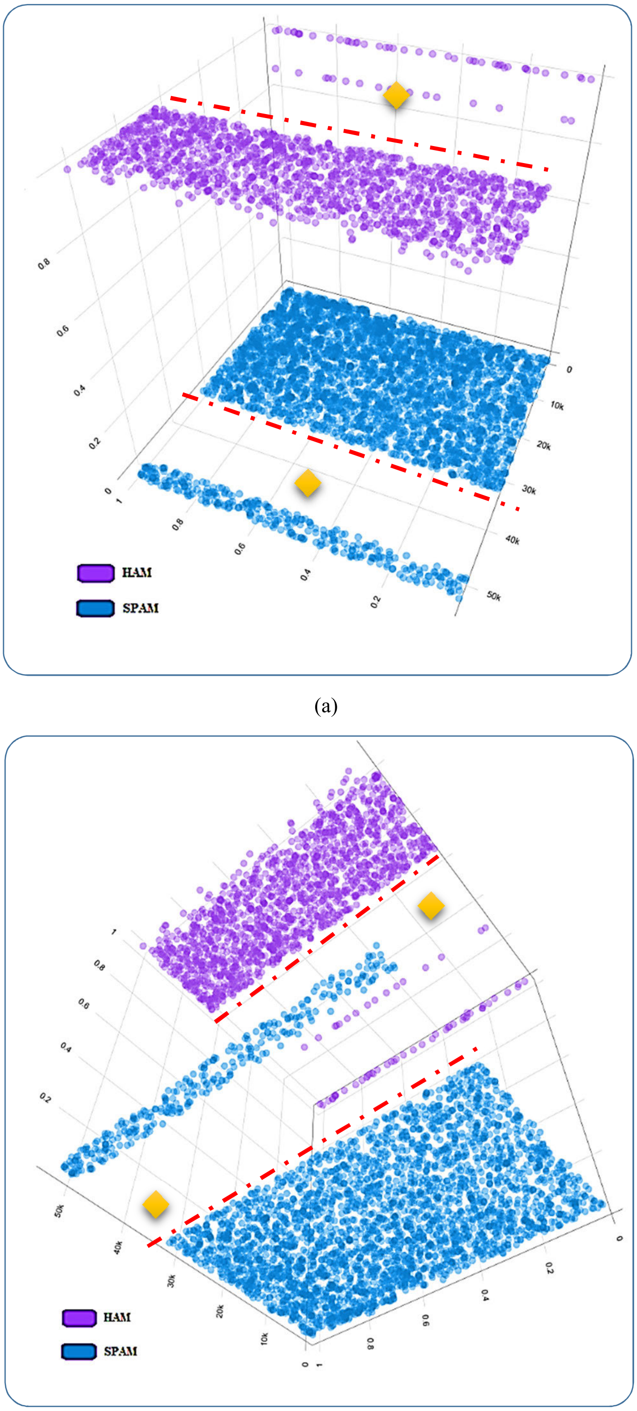
There are noticeably very few misclassifications and overall the cluster structures are closes to the optimum level as illustrated in Fig. 8. Again as there is no direct parameter that takes number of clusters as input, we had to set 'min_cluster_size' to 23% of the data, 'min_sample' to 50 and 'cluster_method' as 'xi' in order to produce a two-cluster solution.

The visualised output of the above discussed clustering algorithms indicate indeed that not all the algorithms are suitable for clustering emails into ham and spam. K-means, OPTICS and Spectral seem to be producing better clusters than BIRCH, K-modes and HDBSCAN. However, we need to quantify the results with proper validation methods to cement a decision with high degree of confidence.

Python's scikit-learn's [49] implementations of the algorithms have been used for this research initiative.

g: CHOICES OF DISTANCE METRICS

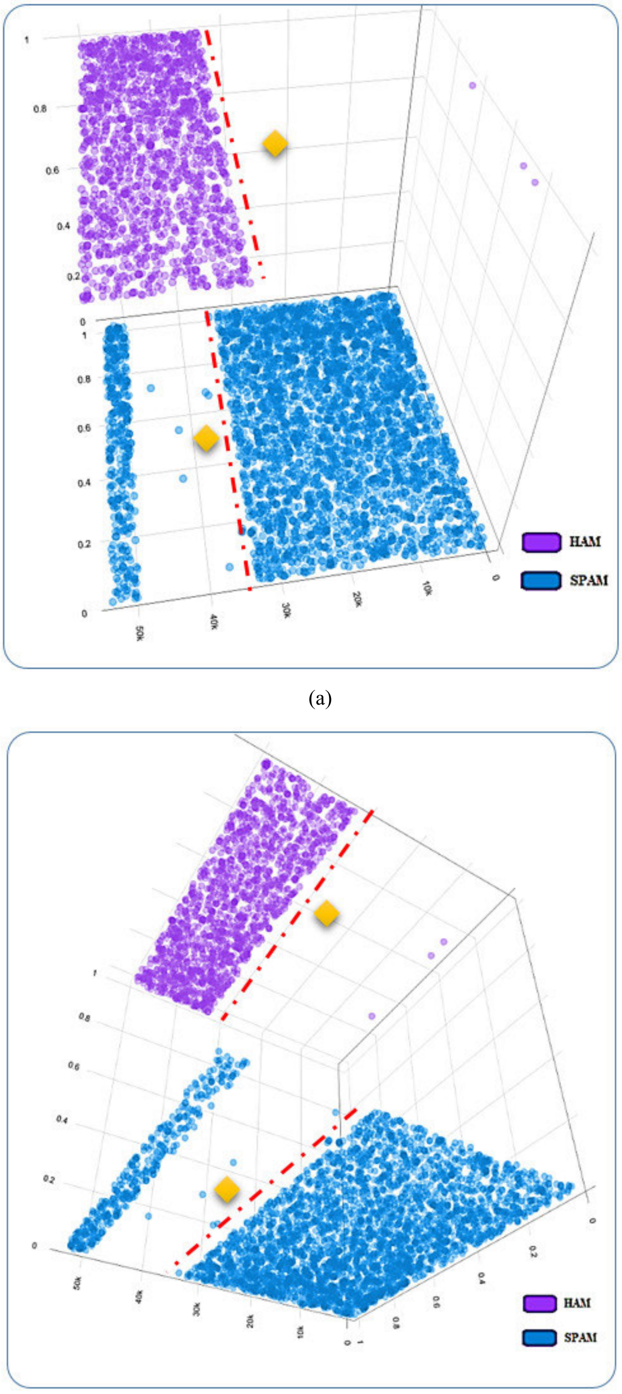
The proper selection of Distance Metrics (signifying how similar or far apart are a pair of data points) is essential to



(a)

(b)

FIGURE 13. **(a). Top-down view of clusters (k-means). (b). Bottom-up view of clusters (k-means).**



(a)

(b)

FIGURE 14. **(a). Top-down view of clusters (OPTICS). (b). Bottom-up view of clusters (OPTICS).**

any clustering and critically affects the performance of the algorithms. Oftentimes it depends on the type of data in the dataset and the problem domain.

Due to the numeric nature of our dataset, the Euclidean distance has been used for all the algorithms, except for Spectral clustering, where the 'rbf' (Radial Basis Function [50]) kernel has been preferred (however, 'rbf' internally uses Euclidean Distance as well) and in K-modes, where the dissimilarity is calculated differently (the technique is often known as 'Hamming Distance [51]') to that of Euclidean distance.

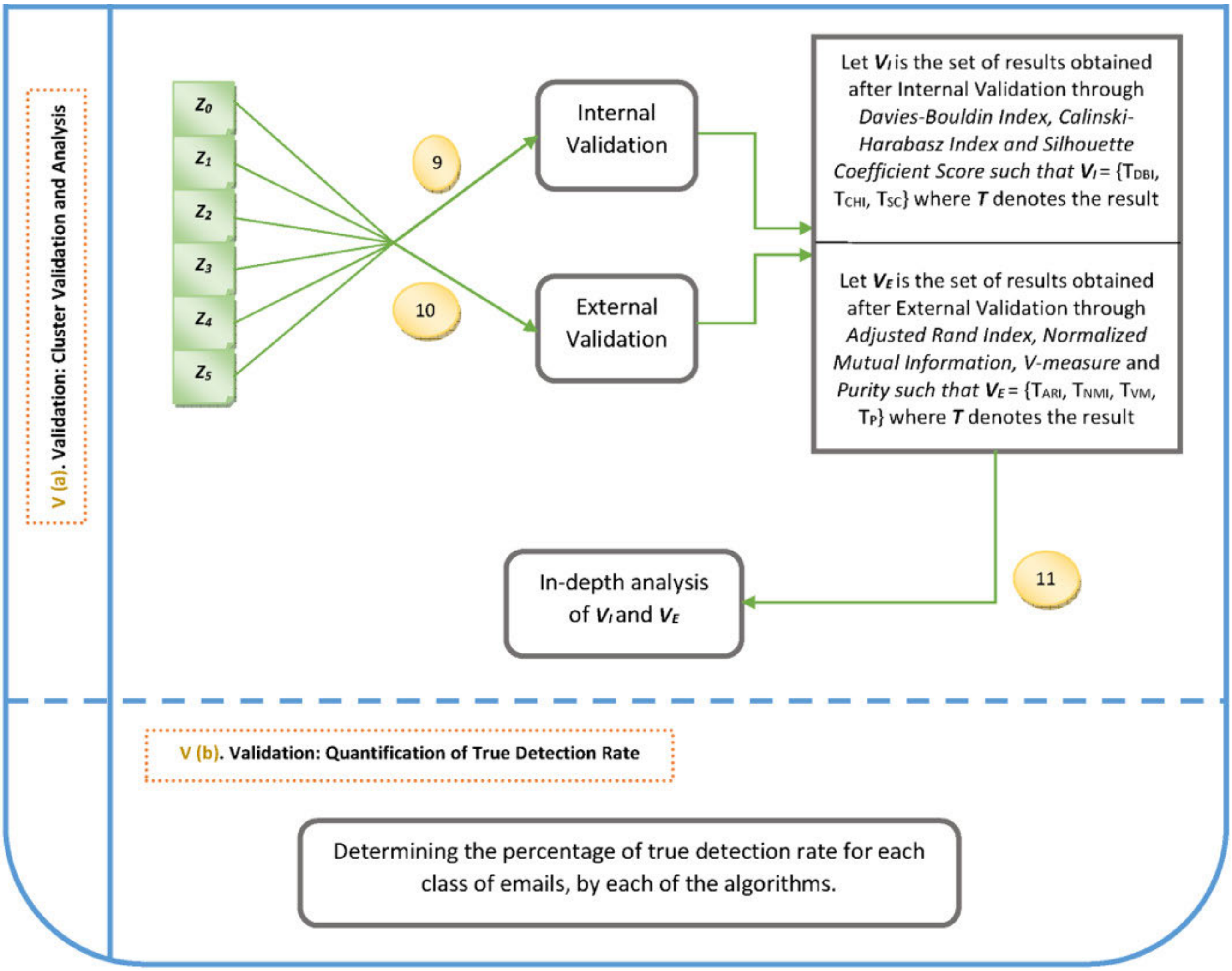

V (a). Validation: Cluster Validation and Analysis
Z_0
Z_1
Z_2
Z_3
Z_4
Z_5
9
10
Internal Validation
External Validation
Let V_I is the set of results obtained after Internal Validation through Davies-Bouldin Index, Calinski-Harabasz Index and Silhouette Coefficient Score such that V_I = {T_{DBI}, T_{CHI}, T_{SC}} where T denotes the result
Let V_E is the set of results obtained after External Validation through Adjusted Rand Index, Normalized Mutual Information, V-measure and Purity such that V_E = {T_{ARI}, T_{NMI}, T_{VM}, T_P} where T denotes the result
11
In-depth analysis of V_I and V_E
V (b). Validation: Quantification of True Detection Rate
Determining the percentage of true detection rate for each class of emails, by each of the algorithms.

FIGURE 15. Application of several validation techniques.

Numeric conversion of categorical data with collision avoidance [52], [53], as in our case, need to be handled with appropriate caution. As because 'ranking' of features is *not* one of our purposes, and we are only looking to *group* data points in relevant clusters based on the calculated distance between those data points, the Euclidean distance can be applied in our case.

E. CLUSTER VALIDATION

To measure the 'Goodness' of the produced clusters and to get an objective insight into the clustering algorithms' merit, an extensive degree of validation methods has been applied as shown in Fig. 15. In those case where the optimum number of clusters is unknown, validation can also provide a credible estimate on that. Validation can be done primarily in two ways:

Internally: Such processes evaluate the connectedness (how well a pair of data points within the same cluster is connected to each other than those are to with other immediate date points placed outside the cluster), and the compactness (how closed are the data points, placed inside the same cluster, to each other) [54]. Internal measures do not require any prior cluster labelling or ground-truths. Acceptable clusters have minimal 'Connectedness' and 'Compactness'.

Externally: These validation techniques gauge the degree to which cluster labels match class labels supplied externally [55]. As our datasets are custom-built, we have the luxury of using External measures; note that these class labels have not been used in any of the processes discussed in previous sections. We will also look at the 'True Rate of Detection' (the 'Recall' measure) for each of the clusters.

A number of validation methods as outlined in Table 6 have been applied.

1) INTERNAL VALIDATION

In this section, we will have a look, using various internal metrics, how the clusters have been validated.

a: DAVIES-BOULDIN INDEX

The metric works on the basis of the ratio of *within-cluster* distances to *between-cluster* distances [56]. Smaller

TABLE 6. Validation methods used.

TYPE	METHOD	CRITERIA
Internal	▪ Davies-Bouldin Index[1] ▪ Calinski-Harabasz Index[2] ▪ Silhouette Coefficient Score[2]	[1] Smaller values indicate better defined clusters [2]A higher score relates to a model with better defined clusters
External	▪ Adjusted Rand Index ▪ Normalised Mutual Information ▪ V-measure ▪ Purity	Closer to 1 is optimum; ≤ 0 is poor

the value, better the clustering. A factor to note is that we have used the *reverse* of Davies-Bouldin Index, (*1- Davies-Bouldin Index*). This will reverse the direction but will make it consistent with other indices used in this research, without affecting the overall outcome. The Davies Bouldin Index (DBI) can be calculated for any value of *n_cluster* ($\boldsymbol{n_c}$) using (14) [63], where d is the Euclidian Distance between the points, $\boldsymbol{c_j}$ is the cluster $\boldsymbol{j}$ having $\boldsymbol{x_j}$ as the centroid,

$$\boldsymbol{DBI} = \frac{\mathbf{1}}{\boldsymbol{n_c}}\sum_{j=1}^{n_c}\max_{k=1..n_c,k\neq j}(\boldsymbol{R_{jk}})$$

$$\textbf{where, } \boldsymbol{R_{jk}} = \frac{\frac{1}{||c_j||}\sum_{y\in c_j} d(y,x_j)+\frac{1}{||c_k||}\sum_{y\in c_k} d(y,x_k)}{d(x_j,x_k)} \tag{14}$$

From Fig. 16 we can observe that the metric considers Spectral, K-means and OPTICS to be of 'almost' equal performers while K-modes is considered the poorest, followed by HDBSCAN and BIRCH. The conclusion slightly defers from the knowledge that we have gained through Scatterplot visualisation of clusters in previous sections, where K-modes had been thought to have edged out both BIRCH and HDB-SCAN, Also, as indicated by the scatterplots, OPTICS, on a visual scale at least, had achieved the optimum clustering, which we can see here as well. However, we will investigate further, through other metrics to determine the degree of correlation to our visual clues.

b: CALINSKI-HARABASZ INDEX

A ratio-type index that evaluates the cluster validity by comparing the average between- and within cluster sum of squares [57]. A higher value indicates better proposition. The index, $\boldsymbol{CH_k}$, is defined as in (15) [64], where $\boldsymbol{V_b}$ is the overall

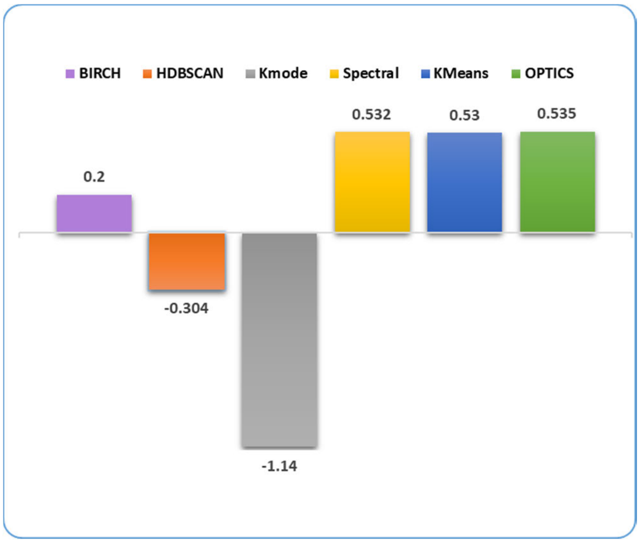


FIGURE 16. Validated outcomes for Davies-Bouldin Index.

between-cluster variance, $\boldsymbol{V_w}$ is the overall within-cluster variance, $\boldsymbol{N}$ is the number of observations and $\boldsymbol{k}$ denotes the total number of clusters.

$$\boldsymbol{CH_k} = \frac{\boldsymbol{V_b}}{\boldsymbol{V_w}}\boldsymbol{x}\frac{\boldsymbol{N}-\mathbf{1}}{\boldsymbol{K}-\mathbf{1}} \tag{15}$$

The Calinski-Harabasz Index does not have any maximum range thus the results returned can be quite long, for instance the measure for Spectral's performance has been quantified as 111546.382. Thus we have confined the results within the range of 0 and 1, which may diminish the differences among algorithms slightly, but still provided the general trend. In addition, we were able to gather comparative insights in relation to other indices.

In this research, a slight variation of Sigmoid function, as shown in (16) has been applied to the outputs of Calinski Harabasz Index to 'squash' it within 0 to 1 so that for each output $\boldsymbol{x}$, we get a corresponding value $\boldsymbol{v}$ after squashing such that $\{\mathbf{v}\text{: } \mathbf{v} > \mathbf{0} \textbf{ and } \mathbf{v} < \mathbf{1}\}$. $\boldsymbol{n}$ is the length of the integer part of the highest value returned by the Calinski-Harabasz Index.

$$f\ (\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1}+e^{-x/n}} \tag{16}$$

Fig. 17 projects the evaluated results which somewhat matches to what we have assumed from the scatterplot cluster visualisation.

Though OPTICS seem to have performed lesser than K-means and Spectral, while all other algorithms showed unsatisfactory clustering, with BIRCH being the poorest. Additionally, in reality the difference between the algorithms that have shown promising results and the rest, is actually quite substantial and sharper than what appears in the figure, but the general trend remains the same.

c: SILHOUETTE COEFFICIENT SCORE

One of the most widely used internal cluster validation techniques. The Silhouette Coefficient score, **c**, is derived for each

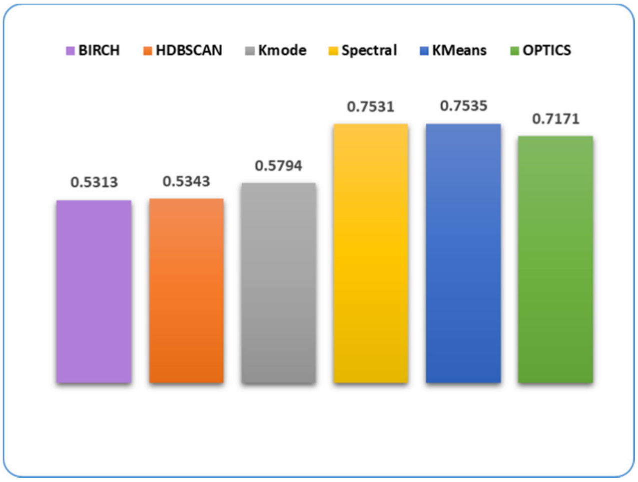


FIGURE 17. **Validated outcomes for Calinski-Harabasz Index.**

of the samples using the mean within-cluster (intra-cluster) distance **p** and the mean nearest-cluster distance **q**, generally using (17) [58].

$$c = \frac{(q-p)}{max(p,q)} \tag{17}$$

where, **q** is the distance between a sample and the nearest cluster that the sample is not a part of. The metric is primarily an intuitive graphical tool that aids the user in visually assessing cluster quality.

The Silhouette Coefficient scores for each of the algorithms have been charted in Fig. 18.

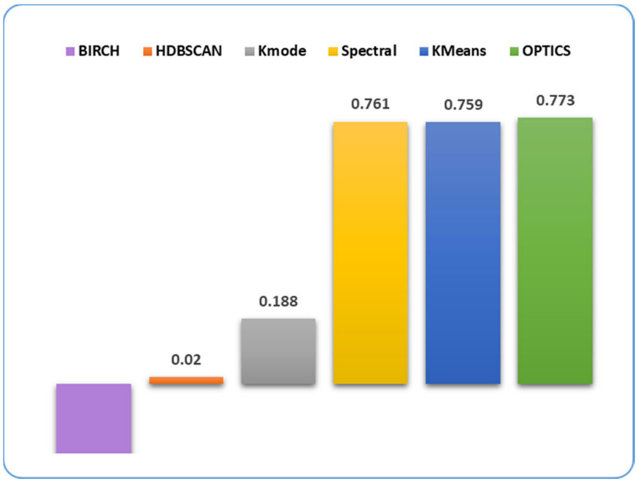


FIGURE 18. **Validated outcomes for Silhouette Coefficient Score.**

The above chart shows maximum resemblance to our assumptions internalised from previously discussed cluster scattepplots. OPTICS clearly performed best with K-means and Spectral are not so far behind, whereas the remaining three projected disproportionately unsatisfactory results.

In Fig. 17 we can see Silhouette Plots for each of the algorithms, derived from a subsample of $\mathbf{X}^{\mathbf{r}}_{\mathbf{P1}}$ (the ratio of spam to ham kept the same in the subsample). Both the clusters are shown in each of the plots. Though there are some outliers, generally Spectral and OPTICS fared better (approaching towards +1), with K-means close behind. The plots give us a quick visual perspective of the clustering quality based on internally calculated Silhouette scores, even though the scores will not exactly match the scores of Fig. 19, as the observations used is a set of limited subsample, but still we can get a general idea and gauge how closely it corresponds to our findings till now. The red segment of the plot represents the cluster of spam emails, while the teal one is for cluster of ham.

d: SUMMARISING THE INTERNAL VALIDATION OUTCOMES

As we have gone through a number of internal validation techniques, Table 7 presents a summarised view of the outcomes, summing up the positions for each of the algorithms across the validation charts; from which it is quite clear that OPTICS and Spectral showed commendable performance while K-means also not so far behind, however, other three algorithms did not have much of a positive clustering outcome. The scatterplots of section V.D.3 also picturised a similar pattern.

However, to get a complete and comprehensive picture, we will now carry out a number of External validations.

2) EXTERNAL VALIDATION

This section will provide a detail inspection, using various external metrics, on the quality of the clustering.

a: ADJUSTED RAND INDEX (ARI)

The Rand Index (RI) works out a similarity measure between two sets of clusterings by taking into account all pairs of provided samples and totaling pairs that are assigned in the same or different clusters in the predicted as well as in the true clusterings [59]. The raw RI score is then 'adjusted for chance' into the ARI using (18).

$$ARI = \frac{RI - Expected\_RI}{max(RI) - Expected\_RI} \tag{18}$$

Scores closer to 1 signify better clustering. Fig. 20 graphically relays the results after validation through ARI.

b: ADJUSTED MUTUAL INFORMATION (AMI)

The Mutual Information (MI) quantifies the degree of information the two clusters in question have in common and often in information theory referred to as 'Correlation Measure'. The MI score is then 'adjusted for chance' to get the AMI [60]. AMI of two clusters S and H, is determined using (19), where **T** is the Entropy.

$$AMI(S,H) = \frac{[MI\ (S,H) - Expected\ (MI\ (S,H))\ ]}{[avg\ (T\ (S)\ ,T\ (H)) - Expected\ (MI\ (S,H))]} \tag{19}$$

Scores closer to 1 signify better clustering. Fig. 21 graphically relays the results after validation through AMI.

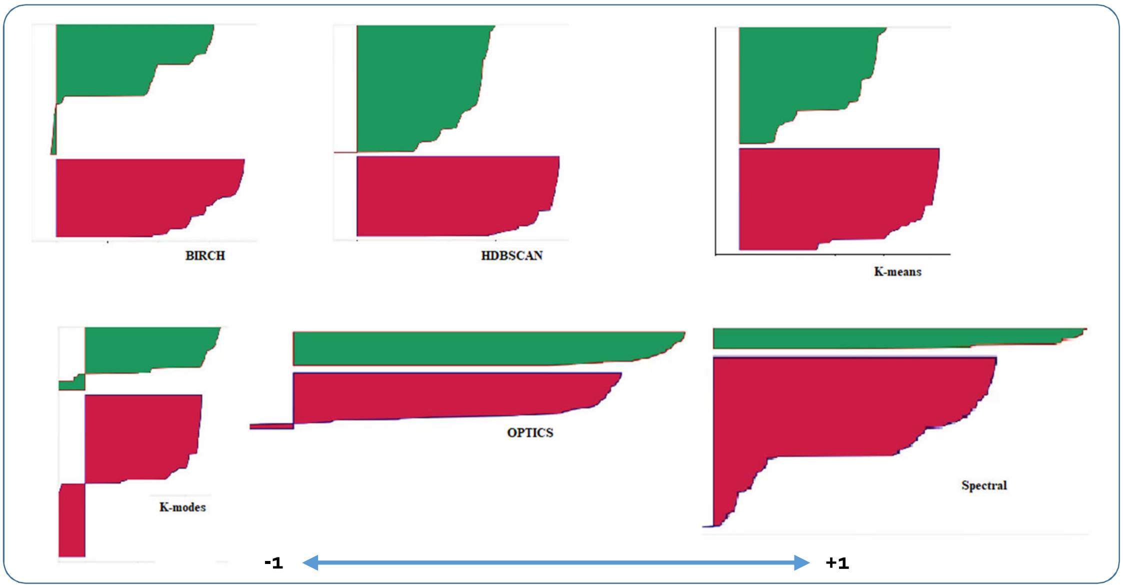


FIGURE 19. Silhouette plots obtained after the clustering attempt of different algorithms on a subsample of X^r_{P1}.

TABLE 7. Summarised view of Internal validation outcomes.

ALGORITHM	∑ POSITION -1	∑ POSITION -2	∑ POSITION -3	∑ POSITION -4	∑ POSITION -5	∑ POSITION -6
BIRCH	-	-	-	1	-	2
HDBSCAN	-	-	-	-	3	-
K-modes	-	-	-	2	-	1
K-means	1	1	1	-	-	-
Spectral	1	2	-	-	-	-
OPTICS	2	-	1	-	-	-

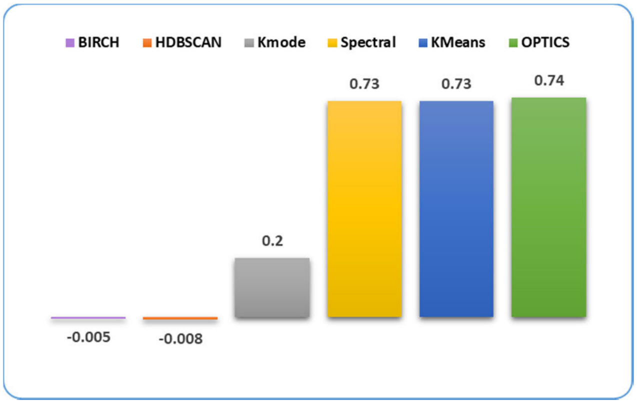


FIGURE 20. Validated outcomes for Adjusted Rand Index.

c: V-MEASURE

V-measure or Validity measure of a cluster is basically a metric developed using conditional entropy analysis. Entropy measures the degree of disorder within a cluster. V-measure takes the Harmonic mean of two important characteristics of a cluster, *Homogeneity* – measure of a cluster holding only members of a single specific cluster, and *Completeness* – whether all members of a given class are allocated to the same cluster [61]. V-measure, $\boldsymbol{v}$ is given in (20). The default value of $\boldsymbol{\beta}$ is 1, signifying equal weightage of homogeneity and completeness.

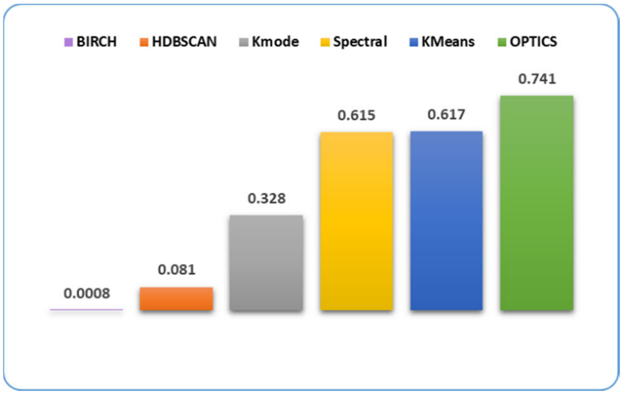


FIGURE 21. Validated outcomes for Adjusted Mutual Information.

$$\boldsymbol{v} = \frac{(\mathbf{1}+\boldsymbol{\beta}) \times \text{homogeneity} \times \text{completeness}}{(\boldsymbol{\beta} \times \text{homogeneity} + \text{completeness})} \tag{20}$$

Scores closer to 1 indicate better clustering. Fig. 22 charts the validation results obtained through V-measure.

d: PURITY

Purity is a simple and transparent external validation measure that is often regarded as the ‘Cluster Accuracy’. Purity is the ratio of the total number of data points belonging to the

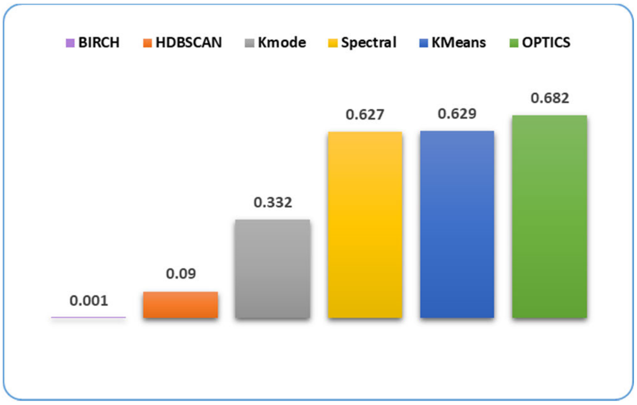


FIGURE 22. Validated outcomes for V-measure.

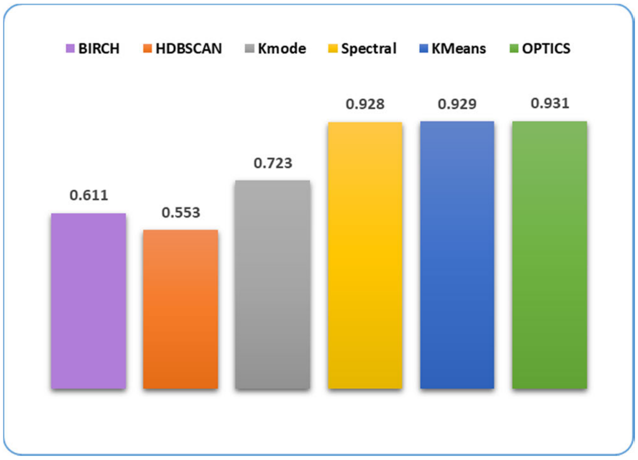


FIGURE 23. Purity of clusters.

TABLE 8. Summarised view of External validation outcomes.

ALGORITHM	∑ POSITION -1	∑ POSITION -2	∑ POSITION -3	∑ POSITION -4	∑ POSITION -5	∑ POSITION -6
BIRCH	-	-	-	-	2	2
HDBSCAN	-	-	-	-	2	2
K-modes	-	-	-	4	-	-
Spectral	-	1	3	-	-	-
K-means	1	3	-	-	-	-
OPTICS	4	-	-	-	-	-

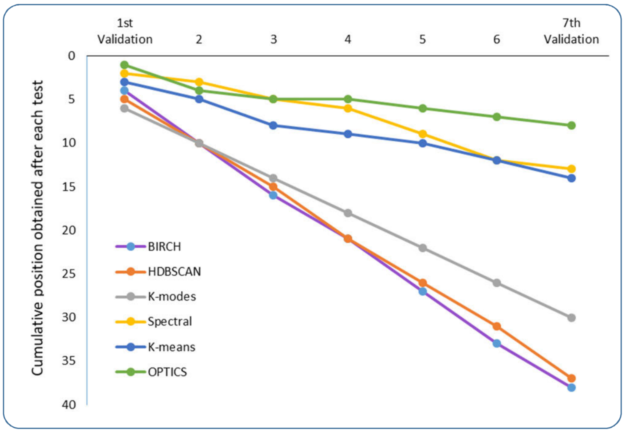


FIGURE 24. Cumulative positions after each validation tests.

dominant class in a cluster to that of its size. Scores closer to 1 suggest better clustering [62]. Fig. 23 shows the measures of purity for each of the algorithms, closely resembling the other measures.

e: SUMMARISING THE EXTERNAL VALIDATION OUTCOMES

As we have gone through a number of external validation techniques, Table 8 presents a summarised view of the outcomes, summing up the positions for each of the algorithms across the validation charts. The results obtained are seemingly consistent with results from internal validations, OPTICS still emerges as the best performer, with K-means and Spectral are perilously close at second spot. The rest of the three algorithms were largely far behind. BIRCH and HDBSCAN were amongst the least performers, whereas K-modes was slightly better, although not satisfactory.

In the next section, cumulative positions for each of the algorithms across all the seven validation tests (Internal and External) will be graphically portrayed.

VI. DETERMINING TOP PERFORMING ALGORITHMS

The figure below (Fig. 24) charts the cumulative positions of each of the clustering algorithms, starting from Davies-Bouldin Index to the Purity measures at 7th spot.

Fig. 24 clearly shows that OPTICS had a clear and uncontested clustering ‘goodness’ throughout; while K-means and Spectral were relatively close for the most part, though Spectral seem to edge ahead with slightly better performance. The rest of the three, as confirmed now, clearly failed to display any commendable rendition and basically way apart from the top three, though K-mode occasionally was moderate in terms of clustering quality.

Thus, in light of the all the validations and clustering scatterplots, we can conveniently say Spectral and K-means have been quite good at producing high quality clusters of Ham and Spam emails, while OPTICS delivered clusters that are closest to the optimum quality, with, on an average, 3.5% better performance, than that of Spectral and K-means ($\boldsymbol{x}$), calculated using (21).

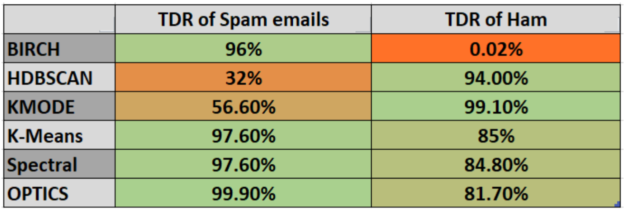
$$\boldsymbol{x} = \frac{\sum_{7}^{n=1} [avg\left(valida_{spec}, valida_{k-mns}\right)]_{validation\_test\_n}}{\mathbf{7}} \tag{21}$$

OPTICS has demonstrated a Purity of 93.1%, while K-means and Spectral scored 92.9% and 92.8% respectively. Such high Purity along with other validation measures, clearly indicate that the quality of ham and spam email clusters produced by these three algorithms based on the header and domain features of emails are quite high.

TRUE DETECTION RATE (THE RECALL MEASURE) OF SPAM AND HAM IN EACH CLUSTER

Well the above external measures indicate the performance of the algorithm as a whole. However, we will now shed lights on another form of external validation- narrowing down to a more granular scale and scrutinise how well each of the algorithms can truly differentiate ham from spam emails by finding out the 'True' detection rate (TDR) of ham and spam emails as indicated in **V(b)** of Fig. 15 (the 'Recall' measure [65] - calculated separately for each clusters).

TABLE 9. True detection rates (TDR).



	TDR of Spam emails	TDR of Ham
BIRCH	96%	0.02%
HDBSCAN	32%	94.00%
KMODE	56.60%	99.10%
K-Means	97.60%	85%
Spectral	97.60%	84.80%
OPTICS	99.90%	81.70%

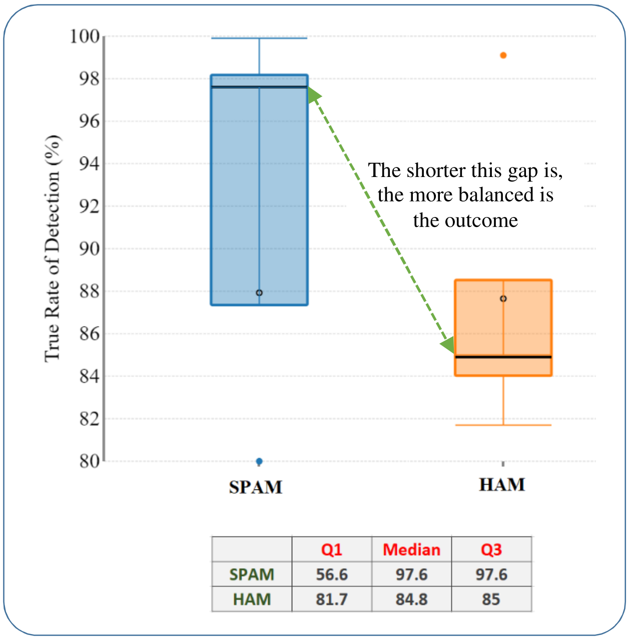
It is quite clear from Table 9 that the optimum algorithms (K-Means, Spectral and OPTICS) have high and balanced true detection rates for both ham and spam emails, while the low performing algorithms may show strength in one specific types of clustering, but not in both; the results for these algorithms may be heavily skewed towards either of the clusters. The distribution of the detection rates can be observed from the Box Plot of Fig. 25. BIRCH and HDBSCAN have not been considered in this Box Plot due to their heavy skewness of detection rates towards certain directions.

Considerable gap between the two median points. This gap should be as minimum as possible for an overall balanced outcome. Additionally, the height of the boxes should be somewhat similar and shorter around the high nineties to account for the balance between ham and spam emails as well as for the high degree of True Detection Rates. The plot analyzes the performance of the algorithms as a group.

VII. EVALUATION OF OUTCOME OF A NEW SET OF DATA ($\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$)

It is important to gauge the performance of these six algorithms on another new set of data as graphically portrayed in Fig. 26, so that the findings above can be confirmed with a considerably high degree of confidence.

The entire process of clustering the raw dataset $\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$ into spam ham and spam emails and validating the results have yielded the Heatmap shown in Fig. 28. The Heatmap itself clearly visualises the clustering quality for Spectral, OPTICS and K-means has been above average under majority of the validation schemes, even in this new dataset, while the other three algorithms have not managed to do well enough, as evidence by the reddish blocks. The finding positively correlates with the outcome of our earlier investigation. K-modes performance, just as before, is not as severe as BIRCH and HDBSCAN, but certainly leaves a lot to desire.



	Q1	Median	Q3
SPAM	56.6	97.6	97.6
HAM	81.7	84.8	85

FIGURE 25. Box plot of true detection rate for both the clusters.

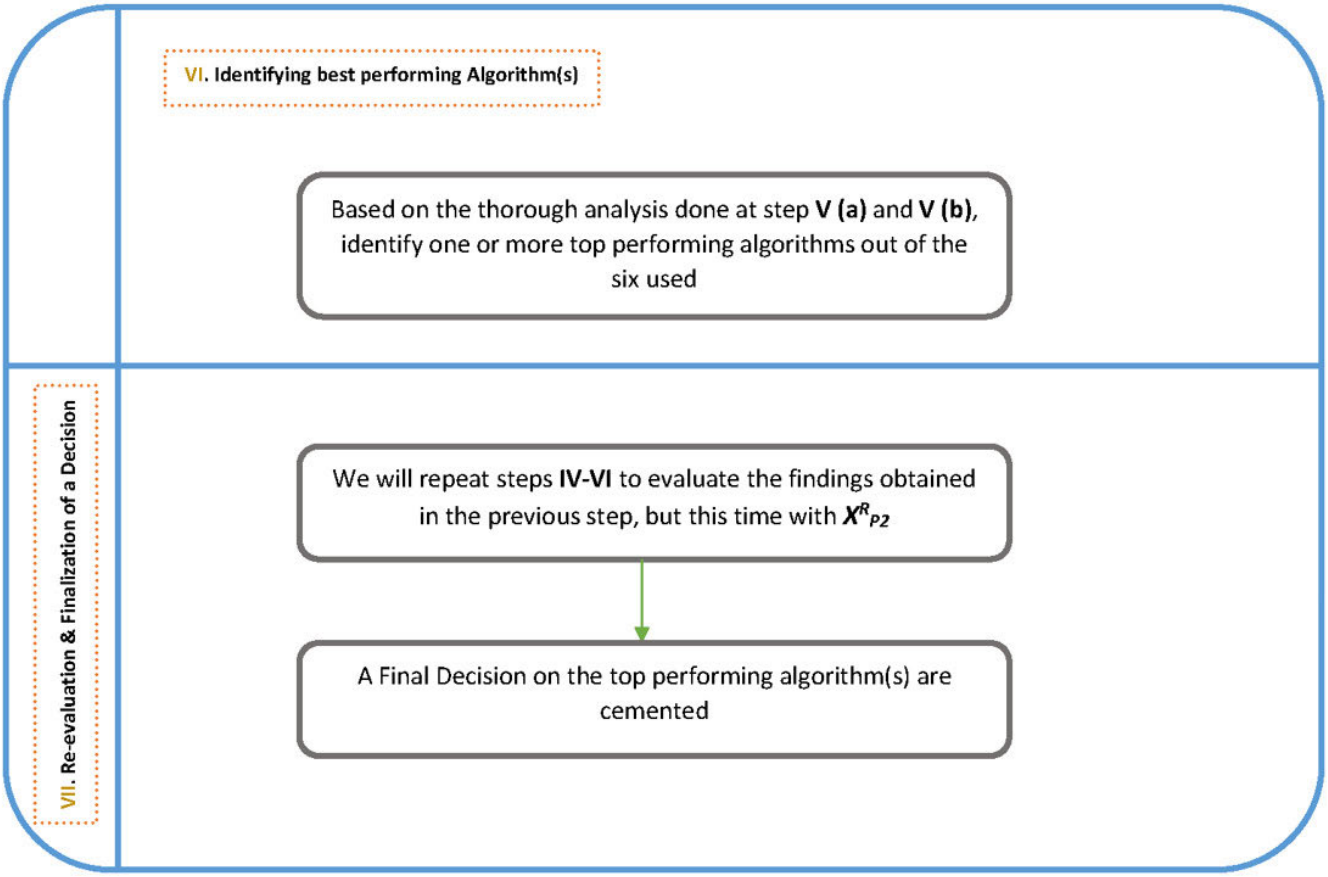
The positions that the algorithms have obtained in each of the validation tests on the clusters for the new dataset, are also quite close to what we had earlier. Fig. 27 presents the cumulative position of each of these algorithms, showing almost a similar trend to that of Fig. 24. It is evident that multiple validation measures need to be executed on the obtained clusters to get a complete picture.

As stated before, Fig. 27 shows the cumulative trends for each of the six algorithms. The figure clearly highlights.

The plot (Fig. 28) clearly projects that in terms of percentages, the concentration of TDR of spam is in high nineties while it is around mid-eighties for TDR of ham (the 'Median' component of the boxes). Thus there is a OPTICS, Spectral and K-means indeed consistently demonstrated strong performances. Spectral seems to have performed slightly better than K-means just as with the other dataset. More importantly, the general trend here projects sharp resemblance to our findings in the previous instance.

Table 10 shows the True Detection Rates for the dataset $\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$. We can clearly observe the similar patterns as well, but the optimum three algorithms this time have demonstrated even better balance between the detection of the class of emails.

Upon deeper scrutiny of distribution patterns of detection rates through Fig. 29, we can discern that the relative difference in detection rates for ham and spam emails is much narrower in this instance than what we had previously, signifying a reasonable balance.


VI. Identifying best performing Algorithm(s)
VII. Re-evaluation & Finalization of a Decision

Based on the thorough analysis done at step **V (a)** and **V (b)**, identify one or more top performing algorithms out of the six used

We will repeat steps **IV-VI** to evaluate the findings obtained in the previous step, but this time with X^R_{P2}

A Final Decision on the top performing algorithm(s) are cemented

FIGURE 26. Identifying top performing algorithms (s) and repeat the clustering process on $\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$.

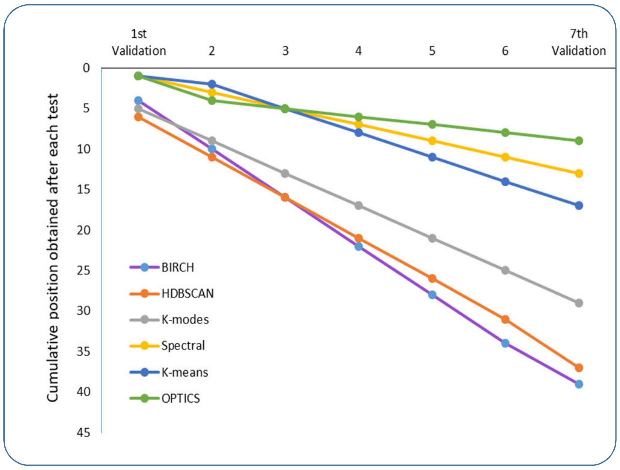


FIGURE 27. Cumulative positions after each validation tests on new dataset.

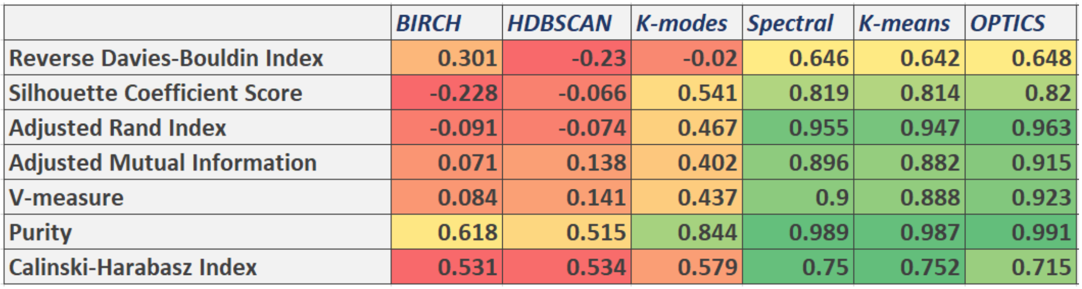
Moreover, the plot suggests overall performance have in fact been better than what we have observed for the dataset $\mathbf{X}^{\mathbf{r}}_{\mathbf{P1}}$ in Fig. 25. BIRCH and HDBSCAN have not been considered in this case due to the considerably poor and skewed outcomes.

TABLE 10. True detection rates (TDR) of $\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$.

	TDR of Spam emails	TDR of Ham
BIRCH	97%	0.019%
HDBSCAN	0.31%	1.00%
KMODE	78.10%	99.50%
K-Means	98.70%	99%
Spectral	98.90%	98.90%
OPTICS	99.80%	97.00%

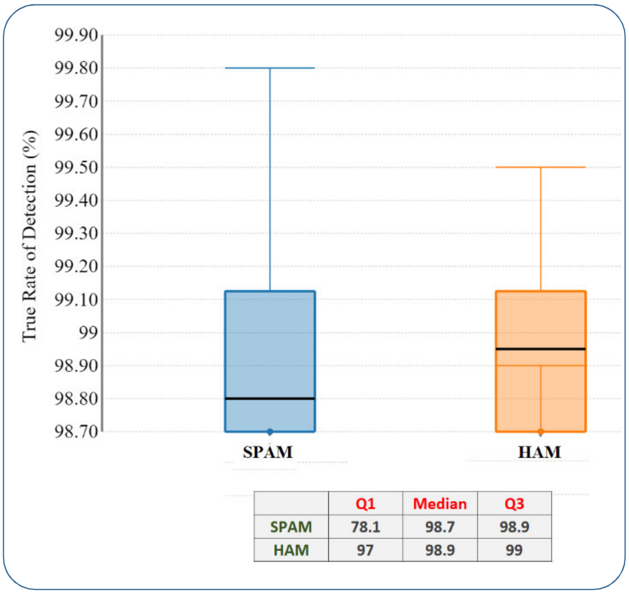
Besides, if we take a look at Fig. 30, it can clearly be observed that the Balanced Accuracy ([TDR of ham + TDR of spam emails] / 2) obtained by the algorithms on $\mathbf{X}^{\mathbf{r}}_{\mathbf{P2}}$ dataset follows the same general trend obtained from $\mathbf{X}^{\mathbf{r}}_{\mathbf{P1}}$. The average balanced accuracy for BIRCH, HDSCAN and K-modes across both the datasets found to be around 64.5%, while in case of OPTICS, Spectral and K-means, it is around 94.91%.

Therefore, in light of this detailed investigation on the performance of six key unsupervised algorithms on clustering ham and spam emails into respective categories, it is fully obvious that OPTICS, Spectral and K-means are able to demonstrate better outcomes than some of the other algorithms, with the margin of difference being quite substantial.



	BIRCH	HDBSCAN	K-modes	Spectral	K-means	OPTICS
Reverse Davies-Bouldin Index	0.301	-0.23	-0.02	0.646	0.642	0.648
Silhouette Coefficient Score	-0.228	-0.066	0.541	0.819	0.814	0.82
Adjusted Rand Index	-0.091	-0.074	0.467	0.955	0.947	0.963
Adjusted Mutual Information	0.071	0.138	0.402	0.896	0.882	0.915
V-measure	0.084	0.141	0.437	0.9	0.888	0.923
Purity	0.618	0.515	0.844	0.989	0.987	0.991
Calinski-Harabasz Index	0.531	0.534	0.579	0.75	0.752	0.715

FIGURE 28. Heatmap of validation outcomes of X^r_{P2}.



	Q1	Median	Q3
SPAM	78.1	98.7	98.9
HAM	97	98.9	99

FIGURE 29. Box plot of true detection rates for both the clusters for the dataset X^r_{P2}.

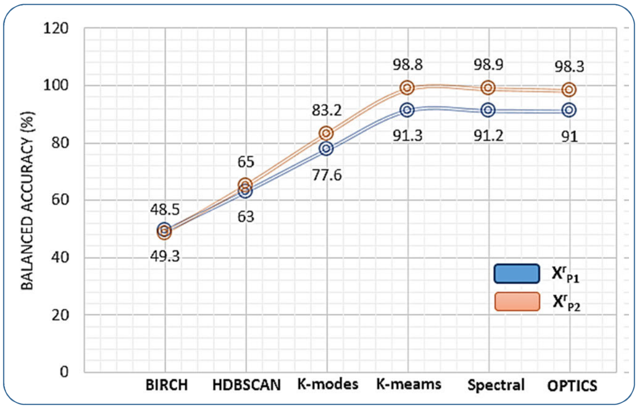


FIGURE 30. Balanced accuracy achieved on both datasets.

Appendix B has the detailed and complete follow diagram of the whole process.

We have compared our work, as shown in Table 11, to a number of the reasonably recent and related initiatives (2014 onward), that use some form of unsupervised learning in the differentiation of ham and spam emails. However, these studies have limitations that can be taken advantage of by the scammers and are not always suitable for implementation in actual business settings. Most of these studies are focused on part of the email framework and do not evaluate a number of important features of both header and content of the emails used for analysis; some of the dataset used for these studies are also quite limited.

VIII. CONCLUSION AND FUTURE WORKS

The paper detailed out the first part of a comprehensive framework based completely on unsupervised methodology to unearth the behavioural pattern in differentiating ham from spam emails through clustering.

Our attempt started off with the creation of the raw database of nearly half a million records of ham and spam emails from multiple email collections; this was a pivotal work that is the basis for not only this research, but also for the forthcoming propositions. From this database we developed a pre-processed dataset of 100,000 records comprising both ham and spam emails, containing critical header and domain information (except for 'subject' field). This dataset was then sliced up into two parts ($\mathbf{X}^r_{P1}$ and $\mathbf{X}^r_{P2}$), containing 60% and 40% of the data respectively, maintaining the same ratio of spam emails to ham. A novel feature reduction method had been applied on the complete dataset before partitioning it to keep the most impactful of header and domain features for clustering purposes. This feature reduction algorithm is an ensemble of three distinct unsupervised feature selection algorithms, namely, PCA, MCFS and Laplacian Feature Selection. The method achieved a 20% reduction of the pre-processed dataset with significantly high confidence.

Afterwards a set of algorithms- OPTICS, Spectral, K-means, HDBSCAN, BIRCH and K-modes were used to cluster the two datasets on two separate runs, and in both cases, of this after thorough validation process, it was found that OPTICS, Spectral, K-means shown commendable performance, while the other three were not optimum. Such a study on identifying ham and spam emails using 'only' unsupervised methods, acting upon solely on email header and domain features (except for 'subject' field) has been a completely novel undertaking.

TABLE 11. A comparison with some relevant studies.

	Approach	Results Reported (Accuracy, %)	Additional Comments\Shortcomings
Chakrabarty *et al.* (2014) [71]	Proposed an amalgamation of unsupervised Minimum Spanning Tree and K-Nearest Neighbour (KNN).	75	The proposition is rigidly tied to the directory structure of individual's email settings and therefore not that flexible. Enhance usability is required to work with different types of email management systems.
Laorden *et al.* (2014) [72]	The proposed techniques utilizes an algorithm known as 'Quality Threshold (QT)', which basically comes under the category of Partitional Clustering algorithms, a close variation of K-Means Clustering.	92.27	The system may perform unexpectedly against the usage of language features such as Synonyms, Hyponyms [73] and Metonymy. Additionally, source based authenticity of the incoming emails have been ignored.
Cabrera-León *et al.* (2016) [74]	Discussed a Self-Organizing-Map (SOM) based system where emails were categorized in 13 different categories using raw Term Frequency measure along with some other metadata.	94.4	The performance of the model against newer and off-topic emails is not consistent. Additionally, no header based measure has been taken.
Smadi *et al.* (2018) [75]	Introduced a system comprising supervised, unsupervised and reinforcement learning based techniques to combat mainly phishing spam attacks.	95.05	The number of features used are extremely high and the authors did not provide significance of using a number of seemingly insignificant features. Besides, common advertising spam emails are not considered.
Martino *et al.* (2020) [11]	A content based multiclass spam email identification framework that uses unsupervised hierarchical clustering along with some supervised classifiers combined with TF-IDF.	95.39	Though it is a multi-class spam detection framework, but the dataset used is heavily skewed towards one particular class, thus a more varied and large-scale testing is required along with an inspection into important header features.
Karim *et al.* (2020)	Approach discussed in this paper.	Average Balanced Accuracy of 94.91	The work only deals with header features at this moment. The subsequent work on other parts of the email will certainly increase its effectiveness.

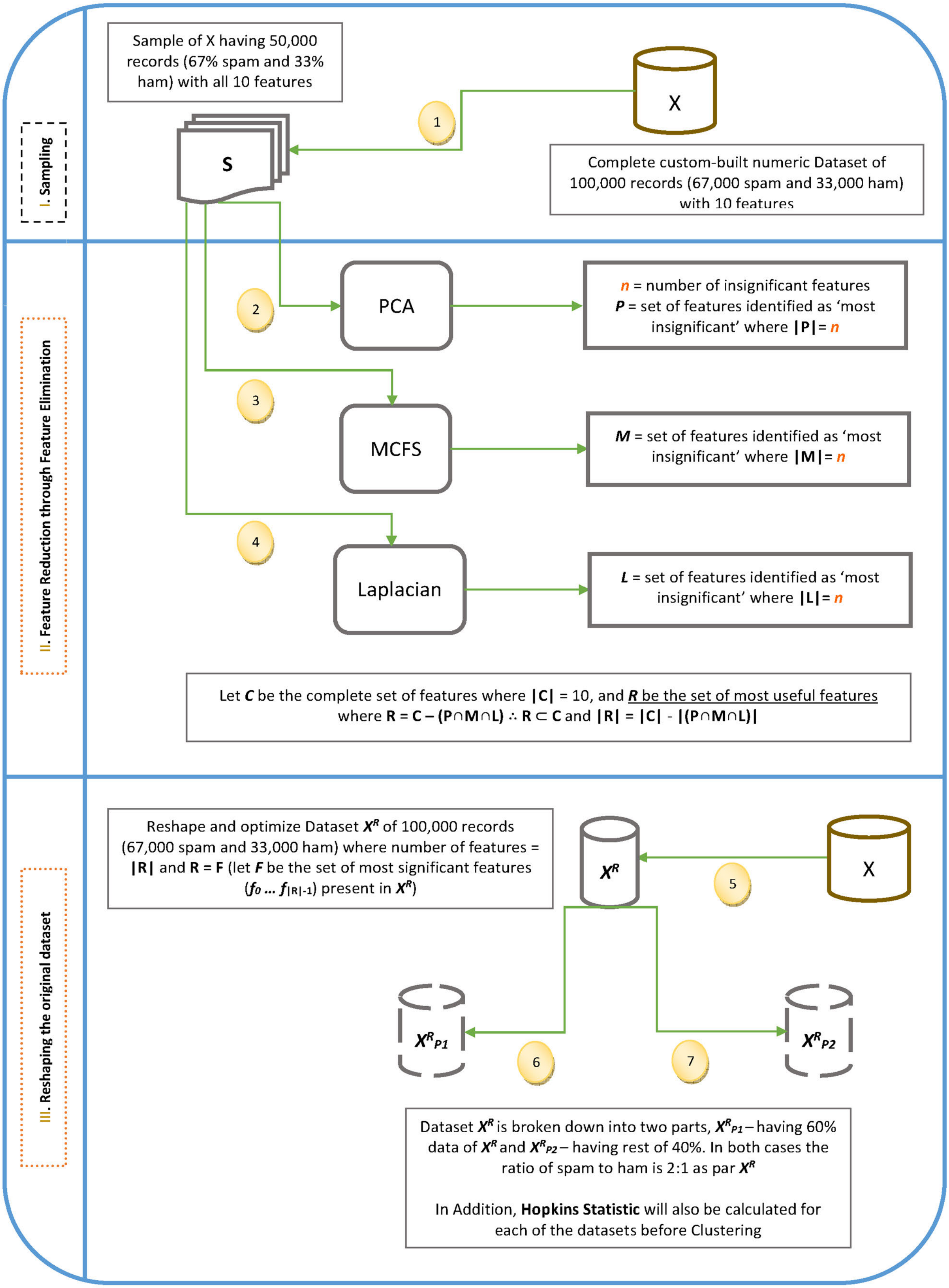
In our future endeavor, we intent to propose the second segment of the framework where algorithms used in this study will be implemented on email body and subject field for clustering purposes. The Purity of the clusters produced by the three best performing algorithms in this study, though is extremely good, but some degree of misclassifications are still there. In future propositions of the framework, we aim to reduce this misclassification rate further, and also implement a third category aside from 'spam' and 'ham', which will cluster 'weakly defined' 'spam' and 'ham' and form a cluster of unspecific data points or emails. In this way the emails identified as spam and ham will have more confidence in its classification, while the users will have the chance to act on this third cluster independently of the system, resulting in a more consistent outcome tailored to users' need, thereby improving user satisfaction and satisfactory balance between all the clusters (minimum skewness). The end results of future research attempts in combination with the knowledge gained from this study, will be the key to develop the intended hybridised unsupervised anti-spam framework.

Additionally, we plan to make our entire database of half a million records publicly available for further research by others when we complete the entire research project through subsequent analysis, experimentation and publication after this article. We have already made the preprocessed dataset of 100,000 records, used for Header analysis in this research, publicly downloadable from *github.com/asif5566/dataextract* for inspection purposes. The future dataset developed for content and subject analysis will also be made available in due course. We believe such a large, dense and ready-made database of ham and spam emails, containing almost all the relevant fields and content of an entire email, will be a significantly useful contribution for future research undertakings.

APPENDIX A

A. MORE ON DATASET CONSTRUCTION AND PREPROCESSING

We have used Python 3. 6 and several related libraries for gleaning out all possible header fields from the text corpuses


I. Sampling
S
X
1
II. Feature Reduction through Feature Elimination
2
PCA
3
MCFS
4
Laplacian
III. Reshaping the original dataset
X^R
X
5
$X^R{}_{P1}$
6
7
$X^R{}_{P2}$

Sample of X having 50,000 records (67% spam and 33% ham) with all 10 features

Complete custom-built numeric Dataset of 100,000 records (67,000 spam and 33,000 ham) with 10 features

n = number of insignificant features
P = set of features identified as 'most insignificant' where **|P| = *n***

M = set of features identified as 'most insignificant' where **|M| = *n***

L = set of features identified as 'most insignificant' where **|L| = *n***

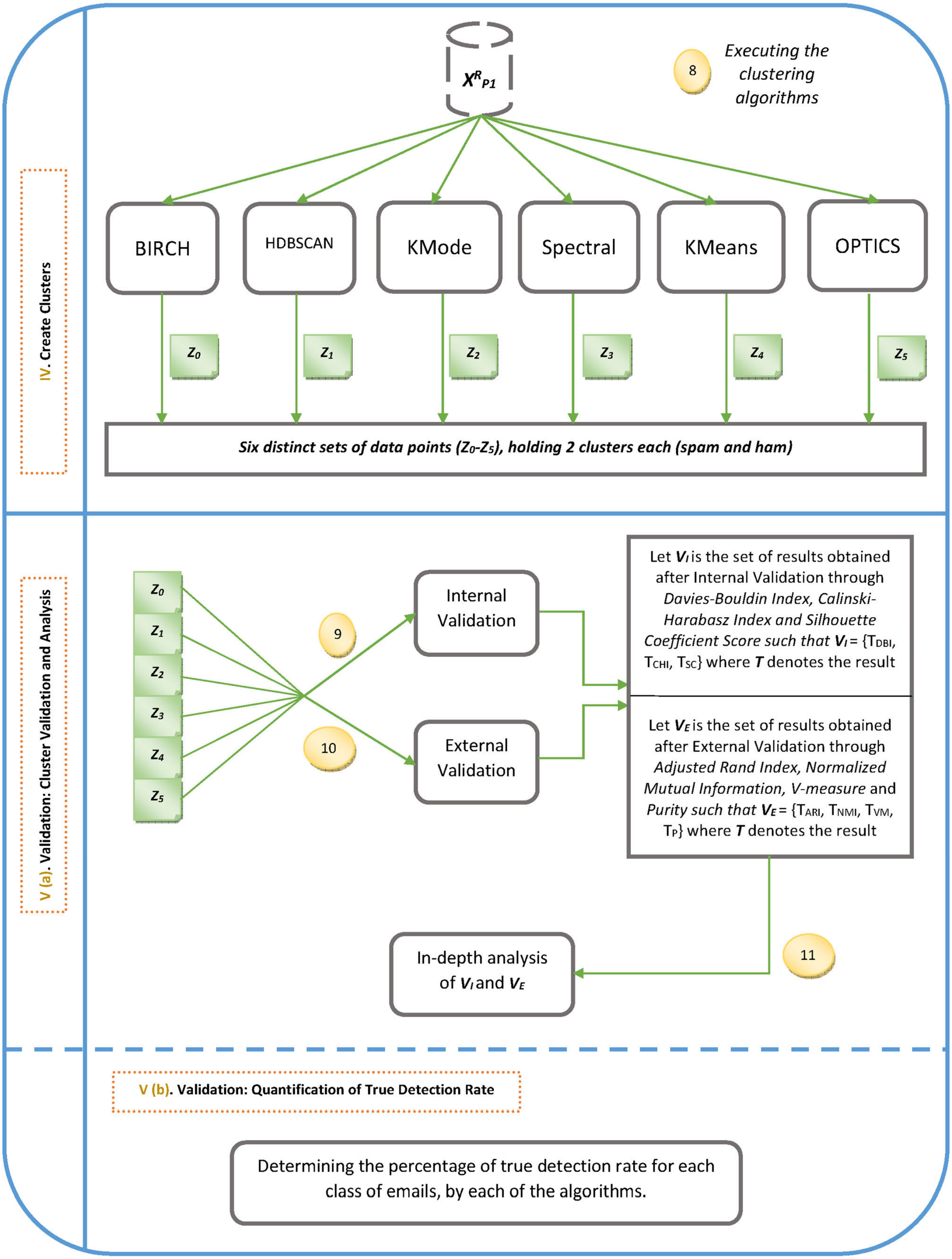
Let ***C*** be the complete set of features where **|C|** = 10, and ***R*** be the set of most useful features where **R = C – (P∩M∩L)** ∴ **R** ⊂ **C** and **|R|** = **|C|** - **|(P∩M∩L)|**

Reshape and optimize Dataset X^R of 100,000 records (67,000 spam and 33,000 ham) where number of features = **|R|** and **R = F** (let ***F*** be the set of most significant features ($f_0 \dots f_{|R|-1}$) present in X^R)

Dataset X^R is broken down into two parts, $X^R{}_{P1}$ – having 60% data of X^R and $X^R{}_{P2}$ – having rest of 40%. In both cases the ratio of spam to ham is 2:1 as par X^R

In Addition, **Hopkins Statistic** will also be calculated for each of the datasets before Clustering

FIGURE 31. ***(Continued.)* Detailed step by step pictorial overview.**


IV. Create Clusters
X^R_{P1}
8
Executing the clustering algorithms
BIRCH
HDBSCAN
KMode
Spectral
KMeans
OPTICS
Z_0
Z_1
Z_2
Z_3
Z_4
Z_5
Six distinct sets of data points (Z_0-Z_5), holding 2 clusters each (spam and ham)
V (a). Validation: Cluster Validation and Analysis
Z_0
Z_1
Z_2
Z_3
Z_4
Z_5
9
10
Internal Validation
External Validation
11
In-depth analysis of V_I and V_E
V (b). Validation: Quantification of True Detection Rate

Let V_I is the set of results obtained after Internal Validation through *Davies-Bouldin Index, Calinski-Harabasz Index and Silhouette Coefficient Score such that* $V_I = \{T_{DBI}, T_{CHI}, T_{SC}\}$ where T denotes the result

Let V_E is the set of results obtained after External Validation through *Adjusted Rand Index, Normalized Mutual Information, V-measure* and *Purity such that* $V_E = \{T_{ARI}, T_{NMI}, T_{VM}, T_P\}$ where T denotes the result

Determining the percentage of true detection rate for each class of emails, by each of the algorithms.

FIGURE 31. ***(Continued.)*** **Detailed step by step pictorial overview.**

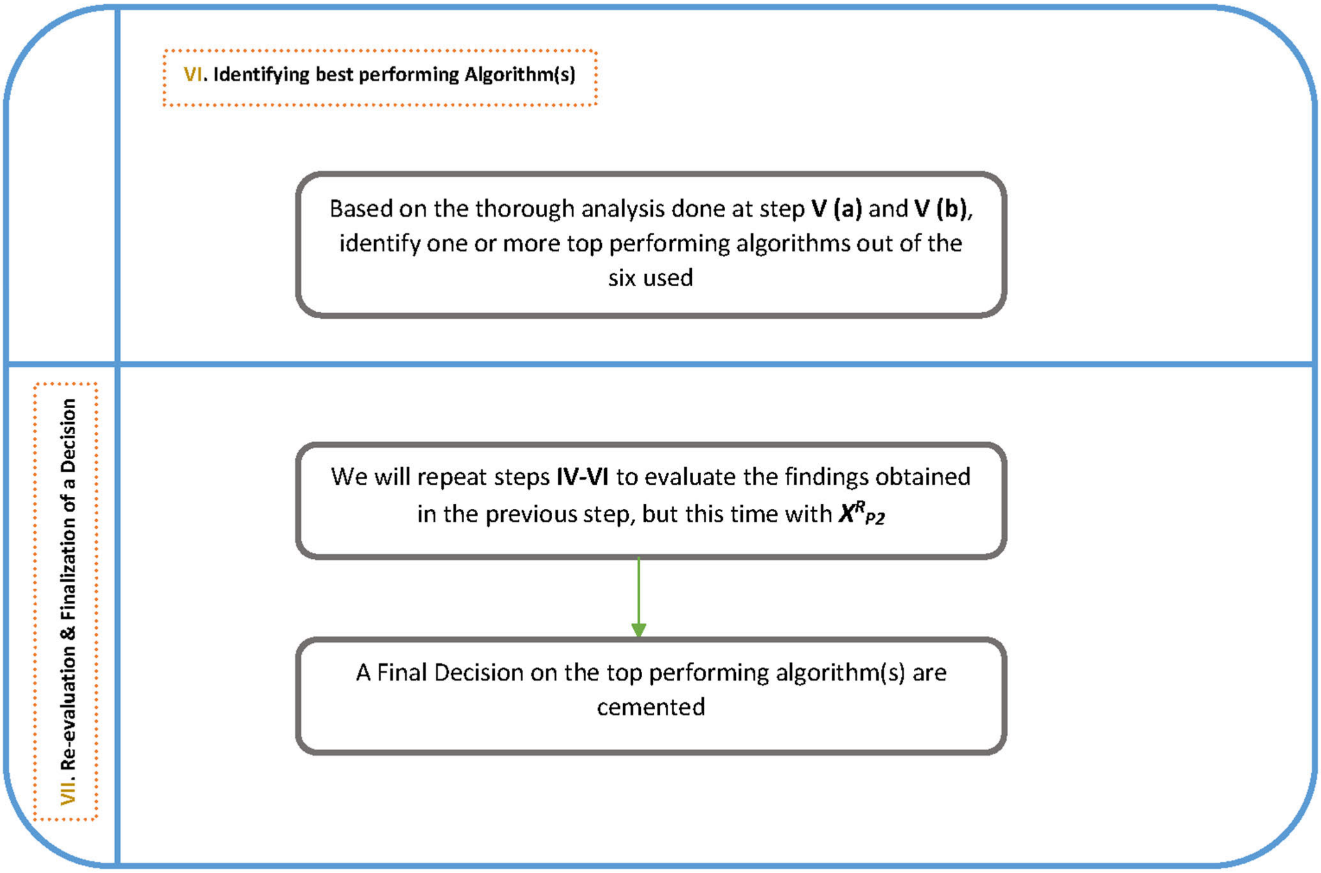

VI. Identifying best performing Algorithm(s)
Based on the thorough analysis done at step **V (a)** and **V (b)**, identify one or more top performing algorithms out of the six used
VII. Re-evaluation & Finalization of a Decision
We will repeat steps **IV-VI** to evaluate the findings obtained in the previous step, but this time with X^R_{P2}
A Final Decision on the top performing algorithm(s) are cemented

FIGURE 31. ***(Continued.)*** **Detailed step by step pictorial overview.**

as well as required preprocessing afterwards. WHOIS records and domain record warehouses have also been consulted for general domain information. A number of mainstream IP Blacklisting databases have been looked into for the status of the source IP. The seminal database of over half a million records has 14 preprocessed header and domain features – from address, date of mail, source IP, whether source IP is blacklisted (Boolean in nature), originating domain, internet domain of the originating country, registrar, age of domain, reply address, return path, message ID, type of the email content, arrival time and total hop count. 'Total hop count' is the total accumulation of the 'Received' field. This database was then used to produce the dataset used in this study. For our subsequent research initiatives, we will be adding the email subject and content to this database. Table 12 lists all the 14 features and the corresponding datatypes and lengths. The datasets used in this research have been derived out of this database as mentioned earlier. Not all the 14 features have been used due to reasons stated earlier as well.

TABLE 12. **Feature description for the seminal database.**

FEATURE	Data Type (MySQL)
From_address	VARCHAR (60)
Source_IP	VARCHAR (20)
Originating_domain	VARCHAR (100)
Originating_country	VARCHAR (10)
Registrar	VARCHAR (60)
Reply_address	VARCHAR (60)
Return_path	VARCHAR (60)
Message_ID	VARCHAR (150)
Content_typ	VARCHAR (15)
Mail_date	DATE
Hop_count	INT(11)
Domain_age	INT(11)
Arrival_time	TIME
Source_IP_blacklisted	CHAR (1)

B. COMPARISON TO SUPERVISED MODELS

Support Vector Machines (SVM) and Naive Bayes (NB) are two most common supervised algorithms using which a number of antispam models have been developed over time, using many common publicly available datasets. To provide a reasonable comparison against such common supervised methods, we trained two supervised models developed using SVM and NB, that showed a 'Test Accuracy' of 97. 44% and 94. 57% with a 60-40 split respectively. The model using SVM achieved somewhat better accuracy than our unsupervised counterpart, while NB performing reasonably at similar scale. With further research down the line, there is a significant possibility that unsupervised model will become considerably more efficient.

APPENDIX B

The complete figure (Fig. 31) showing the whole system architecture at once has been added as part of this Appendix and can be found in the next page.

REFERENCES

[1] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for Spam filtering," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 2, p. 66, Feb. 2012.

[2] E. Bauer. *15 Outrageous Email Spam Statistics that Still Ring True in 2018*. Accessed: Jul. 20, 2019. [Online]. Available: https://www.propellercrm.com/blog/email-spam-statistics

[3] D. Lynkova. *The Surprising Reality of How Many Emails Are Sent Per Day*. Accessed: Jul. 21, 2019. [Online]. Available: https://techjury.net/stats-about/how-many-emails-are-sent-per-day

[4] C. Monitor. *Email Usage Statistics in 2019*. Accessed: Jul. 21, 2019. [Online]. Available: https://www.campaignmonitor.com/blog/email-marketing/2019/07/email-usage-statistics-in-2019/

[5] O. A. Okunade, "Manipulating E-mail server feedback for spam prevention," *Arid Zone J. Eng., Technol. Environ.*, vol. 13, no. 3, pp. 389–397, 2017.

[6] A. Test. *Spam Statistics*. Accessed: Jul. 16, 2019. [Online]. Available: https://www.av-test.org/en/statistics/spam/

[7] R. Islam and Y. Xiang, "Email classification using data reduction method," in *Proc. 5th Int. ICST Conf. Commun. Netw. China*, Aug. 2010, pp. 1–5.

[8] Scamwatch. *Scam Statistics*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.scamwatch.gov.au/about-scamwatch/scam-statistics

[9] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.

[10] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019.

[11] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, Mar. 2018.

[12] K.-L.-A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 1–38, Oct. 2017.

[13] S. Phomkeona, K. Edwards, Y. Ban, and K. Okamura, "Zero-day malicious email behavior investigation and analysis," *Proc. Res. Workshop (APAN)*, 2017, pp. 1–12.

[14] M. Alazab, R. Layton, R. Broadhurst, and B. Bouhours, "Malicious spam emails developments and authorship attribution," in *Proc. 4th Cybercrime Trustworthy Comput. Workshop*, Nov. 2013, pp. 58–68.

[15] S. Dinh, T. Azeb, F. Fortin, D. Mouheb, and M. Debbabi, "Spam campaign detection, analysis, and investigation," *Digit. Invest.*, vol. 12, pp. S12–S21, Mar. 2015.

[16] H. Vyas, "A comparative analysis of frequent pattern mining algorithms," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 5, no. 11, pp. 3010–3012, Nov. 2017.

[17] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, Mar. 2008.

[18] S. A. Al-Saaidah, "Detecting Phishing Emails Using Machine Learning Techniques," M.S. thesis, Dept. Comput. Sci., Middle East Univ., Amman, Jordan, 2017.

[19] A. J. Saleh, A. Karim, B. Shanmugam, S. Azam, K. Kannoorpatti, M. Jonkman, and F. D. Boer, "An intelligent spam detection model based on artificial immune system," *Information*, vol. 10, no. 6, p. 209, Jun. 2019.

[20] R. Chikh and S. Chikhi, "Clustered negative selection algorithm and fruit fly optimization for email spam detection," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 1, pp. 143–152, Jan. 2019.

[21] R. M. Alguliev, R. M. Aliguliyev, and S. A. Nazirova, "Classification of textual E-Mail spam using data mining techniques," *Appl. Comput. Intell. Soft Comput.*, vol. 2011, pp. 1–8, Jan. 2011.

[22] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, p. 367, Jun. 2010.

[23] G. Dougherty, "Unsupervised learning," in *Pattern Recognition and Classification*. New York, NY, USA: Springer Nature, 2012, pp. 143–155.

[24] S. Russell and P. Norvig, "A modern, agent-oriented approach to introductory artificial intelligence," *ACM SIGART Bull.*, vol. 6, no. 2, pp. 24–26, Apr. 1995.

[25] V. Starovoitov, "A clustering technique based on the distance transform," *Pattern Recognit. Lett.*, vol. 17, no. 3, pp. 231–239, Mar. 1996.

[26] C. Si, B. Peng, and X. Li, "PDF-Euclidean distance-based adaptive waveform selection for maximizing radar practical resolution," *IEEE Access*, vol. 7, pp. 148923–148933, 2019.

[27] D. Mallampati, K. C. Shekar, and K. Ravikanth, "Supervised machine learning classifier for Email spam filtering," *Innovations in Computer Science and Engineering* (Lecture Notes in Networks and Systems). Singapore: Springer, 2019, pp. 357–363.

[28] R. M. Mohammad, L. McCluskey, and F. Thabtah, "Intelligent rule-based phishing websites classification," *IET Inf. Secur.*, vol. 8, no. 3, pp. 153–160, May 2014.

[29] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the domain registration behavior of spammers," in *Proc. Conf. Internet Meas. Conf. (IMC)*, 2013, pp. 63–76.

[30] B. Guenter. *Spam Collection*. [Online]. Available: http://untroubled.org/spam/

[31] *TREC Spam Collection*. [Online]. Available: https://trec.nist.gov/data/spam.html

[32] *ENRON Email Corpus*. [Online]. Available: https://www.cs.cmu.edu/~enron/

[33] T. A. Almeida, J. Almeida, and A. Yamakami, "Spam filtering: How the dimensionality reduction affects the accuracy of naive bayes classifiers," *J. Internet Services Appl.*, vol. 1, no. 3, pp. 183–200, Feb. 2011.

[34] C. L. Tan, K. L. Chiew, K. S. C. Yong, S. N. Sze, J. Abdullah, and Y. Sebastian, "A graph-theoretic approach for the detection of phishing webpages," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101793.

[35] X. Kong, C. Hu, and Z. Duan, "Neural networks for principal component analysis," in *Principal Component Analysis Networks and Algorithms*. 2017, pp. 47–73.

[36] I. T. Jolliffe, "Principal components as a small number of interpretable variables: Some examples," in *Principal Component Analysis* (Springer Series in Statistics). New York, NY, USA: Springer Nature, 1986, pp. 50–63.

[37] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 333–342.

[38] X. Zhu, Y. Wang, Y. Li, Y. Tan, G. Wang, and Q. Song, "A new unsupervised feature selection algorithm using similarity-based feature clustering," *Comput. Intell.*, vol. 35, no. 1, pp. 2–22, Feb. 2019.

[39] R. Liu, N. Yang, X. Ding, and L. Ma, "An unsupervised feature selection algorithm: Laplacian score combined with distance-based entropy measure," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, Nov. 2009, pp. 65–68.

[40] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1996, pp-103-114.

[41] J. Sander, "Density-based clustering," in *Encyclopedia of Machine Learning and Data Mining*. New York, NY, USA: Springer, 2016, pp. 1–5.

[42] S. Pettie, "Minimum spanning trees," in *Encyclopedia Algorithms*. New York, NY, USA: Springer, 2016, pp. 1322–1325.

[43] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in *Proc. IOP Conf., Mater. Sci. Eng.*, 2018, vol. 336, no. 1, Art. no. 012017.

[44] V. R. Patel and R. G. Mehta, "Data clustering: Integrating different distance measures with modified k-means algorithm," *Proc. Int. Conf. Soft Comput. Problem Solving*, Dec. 2011, pp. 691–700.

[45] J. Liu and J. Han, "Spectral clustering," in *Data Clustering*. London, U.K.: Chapman and Hall, Mar. 2018, pp. 177–200.

[46] H. Zhou, Y. Zhang, and Y. Liu, "A global-relationship dissimilarity measure for the k-modes clustering algorithm," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–7, Mar. 2017.

[47] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, vol. 99, 1999, pp. 49–60.

[48] G. S. Semaan, A. C. Fadel, J. A. D. M. Brito, and L. S. Ochi, "A hybrid heuristic with hopkins statistic for the automatic clustering problem," *IEEE Latin Amer. Trans.*, vol. 17, no. 01, pp. 7–17, Jan. 2019.

[49] C. Albon, *Machine Learning With Python Cookbook: Practical Solutions From Preprocessing to Deep Learning*. Sebastopol, CA, USA: OReilly, 2018.

[50] J. Liu, "RBF neural network control based on gradient descent algorithm," in *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*. Beijing, China: Springer, 2012, pp. 55–69.

[51] M. Qin, "Hamming-distance-based binary representation of numbers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1–9.

[52] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J. Big Data*, vol. 7, no. 1, Dec. 2020.

[53] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: Binary versus one-hot and feature hashing," M.S. thesis, School Elect. Eng. Comput. Sci., KTH-Roy. Inst. Technol., Stockholm, Sweden, 2018.

[54] L. Jegatha Deborah, R. Baskaran, and A. Kannan, "A survey on internal validity measure for cluster validation," *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, pp. 85–102, Nov. 2010.

[55] S. Gajawada and D. Toshniwal, "Hybrid cluster validation techniques," in *Advances in Computer Science, Engineering & Applications* (Advances in Intelligent Systems and Computing). Berlin, Germany: Springer, 2012, pp. 267–273.

[56] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation," in *Proc. IOP Conf., Mater. Sci. Eng.*, vol. 725, Jan. 2020, Art. no. 012128.

[57] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 911–916.

[58] H. B. Zhou and J. T. Gao, "Automatic method for determining cluster number based on silhouette coefficient," *Adv. Mater. Res.*, vol. 951, pp. 227–230, May 2014.

[59] R. R. de Vargas and B. R. C. Bedregal, "A way to obtain the quality of a partition by adjusted rand index," in *Proc. 2nd Workshop-School Theor. Comput. Sci.*, Oct. 2013, pp. 67–71.

[60] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1–8.

[61] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007.

[62] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2012.

[63] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[64] X. Wang and Y. Xu, "An improved index for clustering validation based on silhouette index and Calinski-Harabasz index," in *Proc. IOP Conf., Mater. Sci. Eng.*, vol. 569, Aug. 2019, Art. no. 052024.

[65] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*. New York, NY, USA: Cambridge Univ. Press, 2014.

[66] A. Zhao, J. Dong, and H. Zhou, "Self-supervised learning from multi-sensor data for sleep recognition," *IEEE Access*, vol. 8, pp. 93907–93921, 2020.

[67] P. Nousi and A. Tefas, "Self-supervised autoencoders for clustering and classification," in *Evolving Systems*. Springer Nature, May 2018, doi: 10.1007/s12530-018-9235-y.

[68] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *Advances in Swarm and Computational Intelligence* (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2015, pp. 3–15.

[69] L. Zhang, Y. Nie, and S. Duan, "Chinese spam filtering based on stacked denoising autoencoders," in *Proc. J. Phys., Conf.*, vol. 1237, 2019, Art. no. 032012.

[70] S. Douzi, M. Amar, and B. El Ouahidi, "Advanced phishing filter using autoencoder and denoising autoencoder," in *Proc. Int. Conf. Big Data Internet Thing (BDIOT)*, 2017, pp. 125–129.

[71] A. Chakrabarty and S. Roy, "An optimized k-NN classifier based on minimum spanning tree for email filtering," in *Proc. 2nd Int. Conf. Bus. Inf. Manage. (ICBIM)*, Jan. 2014, pp. 47–52.

[72] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Inf. Sci.*, vol. 277, pp. 421–444, Sep. 2014.

[73] R. D. Kortum, "Hyperonyms and hyponyms," in *Varieties of Tone*. New York, NY, USA: Palgrave Macmillan, 2013, pp. 178–180.

[74] Y. Cabrera-León, P. G. Báez, and C. P. Suárez-Araujo, "Self-organizing Maps in the design of anti-spam filters a proposal based on thematic categories," in *Proc. 8th Int. Joint Conf. Comput. Intell.*, 2016, pp. 1–12.

[75] F. Jáñez-Martino, E. Fidalgo, S. González-Martínez, and J. Velasco-Mata, "Classification of spam emails through hierarchical clustering and supervised learning," 2020, *arXiv:2005.08773*. [Online]. Available: http://arxiv.org/abs/2005.08773

[76] K. Fragos, "A 2-means clustering technique for unsupervised spam filtering," *Int. J. Sci., Basic Appl. Res.*, vol. 38, no. 1, pp. 114–124, 2018.

ASIF KARIM lives in the Port City of Darwin. He is currently a Ph.D. Researcher with Charles Darwin University, Australia. He is also working towards the development of a robust and advanced e-mail filtering system primarily using machine learning algorithms. He has considerable industry experience in IT, primarily in the field of software engineering. His research interests include machine intelligence and cryptographic communication.

SAMI AZAM (Member, IEEE) is a Leading Researcher and a Lecturer with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He has a number of publications in peer-reviewed journals and international conference proceedings. His research interests include computer vision, signal processing, artificial intelligence, and biomedical engineering.

BHARANIDHARAN SHANMUGAM (Member, IEEE) is currently a Research Intensive Lecturer with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He has a large number of publications in several different journals and conference proceedings. His research interest includes the field of cybersecurity.

KRISHNAN KANNOORPATTI is currently a Research Active Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. In addition of being a Stellar Academic and Innovative Researcher, he also has an extensive experience of working with the government bodies in setting up data privacy policies at national and state level.